RECENT ADVANCES AND THE FUTURE GENERATION OF NEUROINFORMATICS INFRASTRUCTURE

EDITED BY: Xi Cheng, Daniel R. Weinberger, Daniel Marcus, John Van Horn,
Venkata Satyanand Mattay and Qian Luo

PUBLISHED IN: Frontiers in Neuroinformatics

frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88919-677-7
DOI 10.3389/978-2-88919-677-7

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.

Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view.

By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: **researchtopics@frontiersin.org**

RECENT ADVANCES AND THE FUTURE GENERATION OF NEUROINFORMATICS INFRASTRUCTURE

Topic Editors:
Xi Cheng, Lieber Institue for Brain Development, USA
Daniel R. Weinberger, Lieber Institute for Brain Development, USA
Daniel Marcus, Washington University in St. Louis, USA
John Van Horn, University of Southern California, USA
Venkata Satyanand Mattay, Lieber Institute for Brain Development, USA
Qian Luo, Walter Reed Army Research Institute, USA

The huge volume of multi-modal neuroimaging data across different neuroscience communities has posed a daunting challenge to traditional methods of data sharing, data archiving, data processing and data analysis. Neuroinformatics plays a crucial role in creating advanced methodologies and tools for the handling of varied and heterogeneous datasets in order to better understand the structure and function of the brain. These tools and methodologies not only enhance data collection, analysis, integration, interpretation, modeling, and dissemination of data, but also promote data sharing and collaboration.

This Neuroinformatics Research Topic aims to summarize the state-of-art of the current achievements and explores the directions for the future generation of neuroinformatics infrastructure. The publications present solutions for data archiving, data processing and workflow, data mining, and system integration methodologies. Some of the systems presented are large in scale, geographically distributed, and already have a well-established user community. Some discuss opportunities and methodologies that facilitate large-scale parallel data processing tasks under a heterogeneous computational environment.

We wish to stimulate on-going discussions at the level of the neuroinformatics infrastructure including the common challenges, new technologies of maximum benefit, key features of next generation infrastructure, etc. We have asked leading research groups from different research areas of neuroscience/neuroimaging to provide their thoughts on the development of a state of the art and highly-efficient neuroinformatics infrastructure. Such discussions will inspire and help guide the development of a state of the art, highly-efficient neuroinformatics infrastructure.

Citation: Cheng, X., Weinberger, R. D., Marcus, D., Horn, J. V. , Mattay, S. V., Luo, Q., eds. (2015). Recent Advances and the Future Generation of Neuroinformatics Infrastructure. Lausanne: Frontiers Media. doi: 10.3389/978-2-88919-677-7

Table of Contents

frontiers in Neuroinformatics

EDITORIAL
published: 16 June 2015
doi: 10.3389/fninf.2015.00015

Going beyond the current neuroinformatics infrastructure

Xi Cheng[1]*, *Daniel Marcus*[2], *John D. Van Horn*[3], *Qian Luo*[4], *Venkata S. Mattay*[1] and *Daniel R. Weinberger*[1]

[1] Lieber Institute for Brain Development, Johns Hopkins Medical Campus, Baltimore, MD, USA, [2] Department of Radiology, Washington University School of Medicine, St. Louis, MO, USA, [3] Laboratory of Neuro Imaging, Department of Neurology, Institute of Neuroimaging and Informatics, University of Southern California, Los Angeles, CA, USA, [4] Center for Military Psychiatry and Neuroscience Research, Walter Reed Army Research Institute, Silver Spring, MD, USA[†]

Keywords: neuroinformatics, infrastructure, neuroscience, informatics, data sharing

The enormous volume of multi-modal neuroimaging data across different neuroscience research communities poses a daunting challenge to traditional methods of data sharing, data archiving, data processing, and data analysis (Van Horn and Toga, 2014).

Neuroinformatics plays a crucial role in creating advanced methodologies and tools for the handling of varied and heterogeneous datasets in order to better understand the structure and function of the brain. These tools and methodologies not only enhance data collection, analysis, integration, interpretation, modeling, and data dissemination, but also promote data sharing and collaboration (Cox, 1996; Smith et al., 2004; Friston, 2006; Marcus et al., 2007; Dinov et al., 2009; Van Horn and Toga, 2009) which are essential elements for making progress efficiently in this rapidly burgeoning field.

The purpose of this special issue is to use case studies of the state-of-art neuroinformatics infrastructure to anticipate and project future generation systems.

A number of leading research groups from different parts of the world were invited to participate in this research topic. Each of the contributions provided a showcase solution to domain specific challenges we currently face. We will try to review these articles according to the categories of the issues they covered. Some articles covered multiple categories. However, due to the limited space, we only discuss them under one category.

Articles by Bartsch et al. (2014), Goscinski et al. (2014), Haselgrove et al. (2014), King et al. (2014), Marenco et al. (2014), Muehlboeck et al. (2014), Rane et al. (2014), Rautenberg et al. (2014), Sherif et al. (2014), and Wood et al. (2014), present solutions for data archiving and related issues including, how to efficiently collect, store, query, visualize and share large volume neuroimaging data. Some of these systems are large in scale, geographically distributed, and already have a large dataset and a well-established user community.

Beyond neuroimaging, Sobolev et al. (2014) present a data management platform for neurophysiological data, and Mouček et al. (2014), and Tripathy et al. (2014) describe techniques and methodologies for collecting and managing electrophysiological data.

Once the incoming data have been archived, there are many other important issues that need to be addressed.

First, how to visualize the data to meet domain-specific needs is still an open-ended research question. Gutman et al. (2014) present a light framework to visualize DICOM images stored in the Extensible Neuroimaging Archive Toolkit (XNAT). Hänel et al. (2014) describe an application with two designs for the 3D visualization of the human brain.

Second, how to efficiently process huge volumes of datasets is challenging especially when bottom-up explorative data analysis becomes more and more popular. Contributions from Andronache et al. (2013), Da Mota et al. (2014), Dinov et al. (2014), Eklund et al. (2014), Friedel et al. (2014), and Mahmud et al. (2014), discuss opportunities and methodologies that facilitate

OPEN ACCESS

Edited and reviewed by:
Daniel Gardner,
Weill Cornell Medical College, USA

****Correspondence:***
Xi Cheng,
chengxi001@gmail.com

[†] *The opinions or assertions contained herein are the private views of the author, and are not to be construed as official, or as reflecting true views of the Department of the Army or the Department of Defense.*

Received: *28 April 2015*
Accepted: *27 May 2015*
Published: *16 June 2015*

Citation:
Cheng X, Marcus D, Van Horn JD, Luo Q, Mattay VS and Weinberger DR (2015) Going beyond the current neuroinformatics infrastructure. Front. Neuroinform. 9:15. doi: 10.3389/fninf.2015.00015

large-scale parallel data processing tasks under a heterogeneous computational environment.

Third, how to mine the data i.e., how to extract meaningful information from the data, is the most challenging part of all. Liu and Calhoun (2014) provide a review of multivariate analyses approaches in Imaging Genetics. Goh et al. (2014) discuss challenges in neuroinformatics of Traumatic Brain Injury neuroimaging analysis in the context of structural, connectivity, and functional paradigms. The manuscript by Miller et al. (2013) describes novel neuroinformatics technologies at 1 mm anatomical scale based on high-throughput 3D functional and structural imaging technologies of the human brain. Xiang et al. (2014) explored novel data analysis methodologies and platforms for handling large volumes of neuromagnetic data with a very wide range of temporal frequencies. Kauppi et al. (2014), introduce a versatile software package for inter-subject correlation based analyses of fMRI data.

Finally, there are a number of contributions discussing other topics important to the neuroinformatics infrastructure. Zaslavsky et al. (2014) describe a prototype implementation of digital atlasing infrastructure initiated by the International Neuroinformatics Coordinating Facility (INCF). Herrick et al. (2014) showcase how to use dictionary service to extend metadata across XNAT database instances. Sarwate et al. (2014) review the relevant literature on differential privacy, a framework for measuring and tracking privacy loss in these settings, and demonstrate the feasibility of using this framework to calculate statistics on data distributed at many sites while still providing privacy. Das et al. (2014) report a case study on how to foster discussion and communication by using an open-source content management system. Evans and Polavaram (2013) provide a general commentary article in the field of computational models of biologically realistic neuronal networks.

We intend this Special Issue as more than a compendium of current systems. We wish to stimulate on-going discussions at the level of the neuroinformatics infrastructure including: –what are the common challenges the next generation of infrastructure will have to address? –what new technologies will be of maximum benefit? –how will we go beyond the limits of the current generation infrastructure? and –what are the key features next generation infrastructure should implement? Such discussions will inspire and help guide the development of a state of the art, highly-efficient neuroinformatics infrastructure. Such research community wide productive catalytic reactions will be a testament to the worthiness of our efforts in creating this Special Issue.

References

Andronache, A., Rosazza, C., Sattin, D., Leonardi, M., D'Incerti, L., and Minati, L. (2013). Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness. *Front. Neuroinform.*7:16. doi: 10.3389/fninf.2013.00016

Bartsch, H., Thompson, W. K., Jernigan, T. L., and Dale, A. M. (2014). A web-portal for interactive data exploration, visualization, and hypothesis testing. *Front. Neuroinform.* 8:25. doi: 10.3389/fninf.2014.00025

Cox, R. W. (1996). AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput. Biomed. Res.* 29, 162–17310.1006/cbmr.1996.0014

Da Mota, B., Tudoran, R., Costan, A., Varoquaux, G., Brasche, G., Conrod, P., et al. Consortium (2014). Machine learning patterns for neuroimaging-genetic studies in the cloud.*Front. Neuroinform.* 8:31. doi: 10.3389/fninf.2014.00031

Das, S., McCaffrey, P. G., Talkington, M. W. T., Andrews, N. A., Corlosquet, S., Ivinson, A. J., et al. (2014). Pain Research Forum: application of scientific social media frameworks in neuroscience. *Front. Neuroinform.* 8:21. doi: 10.3389/fninf.2014.00021

Dinov, I. D., Petrosyan, P., Liu, Z., Eggert, P., Hobel, S., Vespa, P., et al. (2014). High-throughput neuroimaging-genetics computational infrastructure. *Front. Neuroinform.* 8:41. doi: 10.3389/fninf.2014.00041

Dinov, I. D., Van Horn, J. D., Lozev, K. M., Magsipoc, R., Petrosyan, P., Liu, Z., et al. (2009). Efficient, distributed and interactive neuroimaging data analysis using the LONI *Pipeline*. *Front. Neuroinform.* 3:22. doi: 10.3389/neuro.11.022.2009

Eklund, A., Dufort, P., Villani, M., and LaConte, S. (2014). BROCCOLI: software for fast fMRI analysis on many-core CPUs and GPUs. *Front. Neuroinform.* 8:24. doi: 10.3389/fninf.2014.00024

Evans, R. C., and Polavaram, S. (2013). Growing a garden of neurons. *Front. Neuroinform.* 7:17. doi: 10.3389/fninf.2013.00017

Friedel, M., van Eede, M. C., Pipitone, J., Chakravarty, M. M., and Lerch, J. P. (2014). Pydpiper: a flexible toolkit for constructing novel registration pipelines. *Front. Neuroinform.* 8:67. doi: 10.3389/fninf.2014.00067

Friston, K. J. (2006). *Statistical Parametric Mapping*. London: Academic Press

Goh, S. Y. M., Irimia, A., Torgerson, C. M., and Van Horn, J. D. (2014). Neuroinformatics challenges to the structural, connectomic, functional, and electrophysiological multimodal imaging of human traumatic brain injury.*Front. Neuroinform.* 8:19. doi: 10.3389/fninf.2014.00019

Goscinski, W. J., McIntosh, P., Felzmann, U., Maksimenko, A., Hall, C. J., Gureyev, T., et al. (2014). The multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) high performance computing infrastructure: applications in neuroscience and neuroinformatics research. *Front. Neuroinform.* 8:30. doi: 10.3389/fninf.2014.00030

Gutman, D. A., Dunn, W. D. Jr., Cobb, J., Stoner, R. M., Kalpathy-Cramer, J., and Erickson, B. (2014). Web based tools for visualizing imaging data and development of XNATView, a zero footprint image viewer. *Front. Neuroinform.* 8:53. doi: 10.3389/fninf.2014.00053

Hänel, C., Pieperhoff, P., Hentschel, B., Amunts, K., and Kuhlen, T. (2014). Interactive 3D visualization of structural changes in the brain of a person with corticobasal syndrome. *Front. Neuroinform.* 8:42. doi: 10.3389/fninf.2014.00042

Haselgrove, C., Poline, J.-B., and Kennedy, D. N. (2014). A simple tool for neuroimaging data sharing. *Front. Neuroinform.* 8:52. doi: 10.3389/fninf.2014.00052

Herrick, R., McKay, M., Olsen, T., Horton, W., Florida, M., Moore, C. J., et al. (2014). Data dictionary services in XNAT and the Human Connectome Project. *Front. Neuroinform.* 8:65. doi: 10.3389/fninf.2014.00065

Kauppi, J.-P., Pajula, J., and Tohka, J. (2014). A versatile software package for inter-subject correlation based analyses of fMRI.*Front. Neuroinform.* 8:2. doi: 10.3389/fninf.2014.00002

King, M. D., Wood, D., Miller, B., Kelly, R., Landis, D., Courtney, W., et al. (2014). Automated collection of imaging and phenotypic data to centralized and distributed data repositories. *Front. Neuroinform.* 8:60. doi: 10.3389/fninf.2014.00060

Liu, J., and Calhoun, V. D. (2014). A review of multivariate analyses in imaging genetics. *Front. Neuroinform.* 8:29. doi: 10.3389/fninf.2014.00029

Mahmud, M., Pulizzi, R., Vasilaki, E., and Giugliano, M. (2014). QSpike tools: a generic framework for parallel batch preprocessing of extracellular neuronal signals recorded by substrate microelectrode arrays. *Front. Neuroinform.* 8:26. doi: 10.3389/fninf.2014.00026

Marcus, D. S., Olsen, T. R., Ramaratnam, M., and Buckner, R. L. (2007). The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34. doi: 10.1385/NI:5:1:11

Marenco, L. N., Wang, R., Bandrowski, A. E., Grethe, J. S., Shepherd, G. M., and Miller, P. L. (2014). Extending the NIF DISCO framework to automate complex workflow: coordinating the harvest and integration of data from diverse neuroscience information resources.*Front. Neuroinform.* 8:58. doi: 10.3389/fninf.2014.00058

Miller, M. I., Faria, A. V., Oishi, K., and Mori, S. (2013). High-throughput neuroimaging informatics. *Front. Neuroinform*. 7:31. doi: 10.3389/fninf.2013.00031

Mouček, R., Ježek, P., Vařeka, L., Řondík, T., Brůha P, Papež V., Mautner, P., et al. (2014). Software and hardware infrastructure for research in electrophysiology. *Front. Neuroinform*. 8:20. doi: 10.3389/fninf.2014.00020

Muehlboeck, J.-S., Westman, E., and Simmons, A. (2014). TheHiveDB image data management and analysis framework. *Front. Neuroinform.* 7:49. doi: 10.3389/fninf.2013.00049

Rane, P., Haselgrove, C., Hodge, S. M., Frazier, J. A., and Kennedy, D. N. (2014). Structure-centered portal for child psychiatry research. *Front. Neuroinform*. 8:47. doi: 10.3389/fninf.2014.00047

Rautenberg, P. L., Kumaraswamy, A., Tejero-Cantero, A., Doblander, C., Norouzian, M. R., Kai, K., et al. (2014). NeuronDepot: keeping your colleagues in sync by combining modern cloud storage services, the local file system, and simple web applications. *Front. Neuroinform*. 8:55. doi: 10.3389/fninf.2014.00055

Sarwate, A. D., Plis, S. M., Turner, J. A., Arbabshirani, M. R., and Calhoun, V. D. (2014). Sharing privacy-sensitive access to neuroimaging and genetics data: a review and preliminary validation. *Front. Neuroinform*. 8:35. doi: 10.3389/fninf.2014.00035

Sherif, T., Rioux, P., Rousseau, M.-E., Kassis, N., Beck, N., Adalat, R., et al. (2014). CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research. *Front. Neuroinform*. 8:54. doi: 10.3389/fninf.2014.00054

Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E., Johansen-Berg, H., et al. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage* 23(Suppl. 1), S208–S219. doi: 10.1016/j.neuroimage.2004.07.051

Sobolev, A., Stoewer, A., Leonhardt, A., Rautenberg, P. L., Kellner, C. J., Garbers, C., et al. (2014). Integrated platform and API for electrophysiological data. *Front. Neuroinform*. 8:32. doi: 10.3389/fninf.2014.00032

Tripathy, S. J., Savitskaya, J., Burton, S. D., Urban, N. N., and Gerkin, R. C. (2014). NeuroElectro: a window to the world's neuron electrophysiology data. *Front. Neuroinform*. 8:40. doi: 10.3389/fninf.2014.00040

Van Horn, J. D., and Toga, A. W. (2009). Neuroimaging workflow design and data-mining: a Frontiers in neuroinformatics special issue. *Front Neuroinform.* 3:31. doi: 10.3389/neuro.11.031.2009

Van Horn, J. D., and Toga, A. W. (2014). Human neuroimaging as a "Big Data" science. *Brain Imaging Behav*. 8, 323–331. doi: 10.1007/s11682-013-9255-y

Wood, D., King, M., Landis, D., Courtney, W., Wang, R., Kelly, R., et al. (2014). Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools. *Front. Neuroinform*. 8:71. doi: 10.3389/fninf.2014.00071

Xiang, J., Luo, Q., Kotecha, R., Korman, A., Zhang, F., Luo, H., et al. (2014). Accumulated source imaging of brain activity with both low and high-frequency neuromagnetic signals. *Front. Neuroinform*. 8:57. doi: 10.3389/fninf.2014.00057

Zaslavsky, I., Baldock, R. A., and Boline, J. (2014). Cyberinfrastructure for the digital brain: spatial standards for integrating rodent brain atlases. *Front. Neuroinform*. 8:74. doi: 10.3389/fninf.2014.00074

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

TECHNOLOGY REPORT ARTICLE
published: 26 March 2014
doi: 10.3389/fninf.2014.00025

A web-portal for interactive data exploration, visualization, and hypothesis testing

Hauke Bartsch[1]*, Wesley K. Thompson[1], Terry L. Jernigan[2] and Anders M. Dale[1]

[1] Multi-Modal Imaging Laboratory, Department of Radiology, University of California, San Diego, San Diego, CA, USA
[2] Departments of Cognitive Science, Psychiatry, and Radiology, Center for Human Development at University of California, San Diego, San Diego, CA, USA

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Richard C. Gerkin, Carnegie Mellon University, USA
Satya Sahoo, Case Western Reserve University, USA
Paula Sanz Leon, Aix Marseille Université, France

***Correspondence:**
Hauke Bartsch, Multi-Modal Imaging Laboratory, Department of Radiology, University of California, San Diego, 9500 Gilman Drive, San Diego, La Jolla, CA 92093, USA
e-mail: hbartsch@ucsd.edu

Clinical research studies generate data that need to be shared and statistically analyzed by their participating institutions. The distributed nature of research and the different domains involved present major challenges to data sharing, exploration, and visualization. The Data Portal infrastructure was developed to support ongoing research in the areas of neurocognition, imaging, and genetics. Researchers benefit from the integration of data sources across domains, the explicit representation of knowledge from domain experts, and user interfaces providing convenient access to project specific data resources and algorithms. The system provides an interactive approach to statistical analysis, data mining, and hypothesis testing over the lifetime of a study and fulfills a mandate of public sharing by integrating data sharing into a system built for active data exploration. The web-based platform removes barriers for research and supports the ongoing exploration of data.

Keywords: data exploration, data sharing, genetics, data dictionary, imaging, hypothesis testing

1. INTRODUCTION

Data exploration is an interactive approach involving extraction of relevant characteristics from complex datasets with the aim of formulating hypotheses that lead to collection of new data and experiments (Tukey, 1980). In order to shorten the time required for producing and confirming novel results the interactive component of data exploration can be implemented as a frequent switching between phases of data exploration for the purpose of generating hypotheses and hypothesis testing. However, without proper statistical tools that implement appropriate tests and control for multiple comparisons, data exploration can easily degrade into data fishing, with poor reproducibility of hypothesis test results in independent samples.

Data exploration can also be useful for data curation, quality control, guidance, and early intervention if applied during the data acquisition phase of a project. Thus, effective data exploration tools can improve data quality by identifying problems of study design or execution in a timely fashion. Furthermore, data exploration tools can facilitate analyses by abstracting them from technical considerations such as data location, how information is encoded and what file formats are used. Diverse data sources such as demographic, neurocognitive, imaging, and genetic information can be analyzed in a unified manner by implementing guidelines for the selection of appropriate statistical models. Providing a system that actively supports data exploration combined with hypothesis testing across data modalities is a valuable adjunct to facilities focused primarily on data sharing like the Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC) (Buccigrossi et al., 2008) and the database of Genotypes and Phenotypes (dbGaP) (Mailman et al., 2007).

1.1. DATA SOURCES

Medical imaging studies collect anatomical and functional volumetric images in search of biomarkers to detect disease or to characterize normal development. Because the pictorial representation of structures in imaging does not easily lend itself to statistical analysis (unstructured data), the imaging data are processed, usually automatically, resulting in structured data with an organization into quantitative measurements for features in regions of interest (Dale et al., 1999; Desikan et al., 2006; Hagler et al., 2009). If image data are acquired by multiple sites each device might introduce systematic variation in the data that can hinder the detection of effects or introduce spurious correlations. Documenting auxiliary measures such as the identity of the imaging scanner (i.e., device serial number) and the version of the software used to perform image reconstruction provides essential additional information that can lead to increased power and accuracy in statistical analyses.

Demographic information, *neuromedical history*, and *self-report* measures are all captured by questionnaires and digitized in tabular form which results in a mixture of categorical variables and continuous variables like gender, age, or household income. Information about patient and family history and socioeconomic factors provide important context for the interpretation of data from other sources and are often correlated with clinical outcomes (Monzalvo et al., 2012).

Neurological function and *behavior* are measured by tests of cognition, emotion, motor function, and sensory function. Standardized tests to obtain these measurements are available (Wechsler, 2004; Weintraub et al., 2013) and can be used to obtain either raw or age-normalized scores.

High density gene chips measure variation in single nucleotide polymorphisms (SNPs) in a large number of locations across the whole genome. Typically, on the order of 0.5–2.5 million locations are genotyped for each participant (1000 Genomes, 2012; Fjell et al., 2012). Each location is coded as one of several (two or more) alleles for each study participant. Differences in the frequencies of alleles can be linked to behavioral or structural phenotypes. The distribution of alleles can also be compared to known reference populations, providing information about the genetic ancestry mixture for each participant in the study.

Structured data from these different modalities need to be combined for appropriate statistical analyses that can also control for measured covariates. For example, genetic ancestry may covary with imaging measurements (Biffi et al., 2010), or socio-economic data may covary with cognitive measurements (Hurst et al., 2013). When this information is integrated into the statistical analysis, ancestry admixture effects can be disassociated from effects driven by socio-economic factors.

1.2. APPLICATIONS

Many open-source projects and commercial applications provide support for data acquisition and study control (Wang et al., 2008; Harris et al., 2009; OpenClinica LLC., and collaborators), storage and sharing of imaging data (Marcus et al., 2007; DCM4CHEE, 2013), viewing collections of images (Rosset et al., 2004; Weasis, 2013), organizing collections of genetic information (Purcell et al., 2007), statistical analysis of cognitive, self-report and psychophysical measurements (JMP, 1989–2007; R Core Team, 2013). One notable difference between these applications and the application presented here (Data Portal) is that the Data Portal promotes integrated data exploration and statistical analyses across behavioral, imaging, and genetics domains.

In this paper we describe the features of the data portal for exploratory data analysis and hypothesis driven statistical analysis in the context of the Pediatric Imaging, Neurocognition and Genetics (PING) project (Fjell et al., 2012, http://pingstudy.ucsd.edu, see **Figure 1**). The PING study contains information from over 1500 subjects between the ages of 3 and 20 years and was created to provide a publicly shared database able to link genetic information and behavioral measures with developing patterns of brain structural connectivity and morphology.

The Data Portal allows for the simultaneous exploration of roughly 2300 distinct morphological, demographic, and behavioral measures as well as 500,000 genetic measures obtained on each study participant. Investigators can define and execute statistical models online for data exploration and hypothesis testing. This makes it possible to discover and explore patterns in multiple data domains while controlling for covariates using a rigorous statistical framework. For a given statistical model the portal also supports the exploration of multi-modal image data for any individual subject. The displayed data include structural magnetic resonance images (MRI), diffusion tensor images (DTI) such as fractional anisotropy (FA), apparent diffusion coefficients (ADC), and directionally encoded color (DEC) images, atlas based fiber tracks and surface reconstructions for vertex (surface point) based measures for cortical thickness, regional surface area expansion and regional volume expansion. The combination of study-level analysis with the capabilities of personalized, participant specific exploration of key developmental features support data exploration efforts especially during the data acquisition phase of a project.

2. MATERIALS AND METHODS

2.1. NOMENCLATURE

The data portal distinguishes between *projects* as collections of data and *applications* as project neutral entities for data analysis and visualization. This separation supports several projects hosted side-by-side on the same system. Applications use access to project data to implement specific workflows. As an example, the table application can be used to review the registration of diffusion weighted images and structural scans for a large number of subjects. Images are displayed in a table with selected demographic entries for each session. Two example images are displayed in separate columns for each of the structural scans (horizontal section of T1) and the registered diffusion weighted scans (horizontal section of FA). This arrangement of subject information visually highlights any misalignment of images as disagreement of structural information displayed in the two image modalities. A link guides the user to the image viewing application that provides a multi-planar reconstruction of available image volumes.

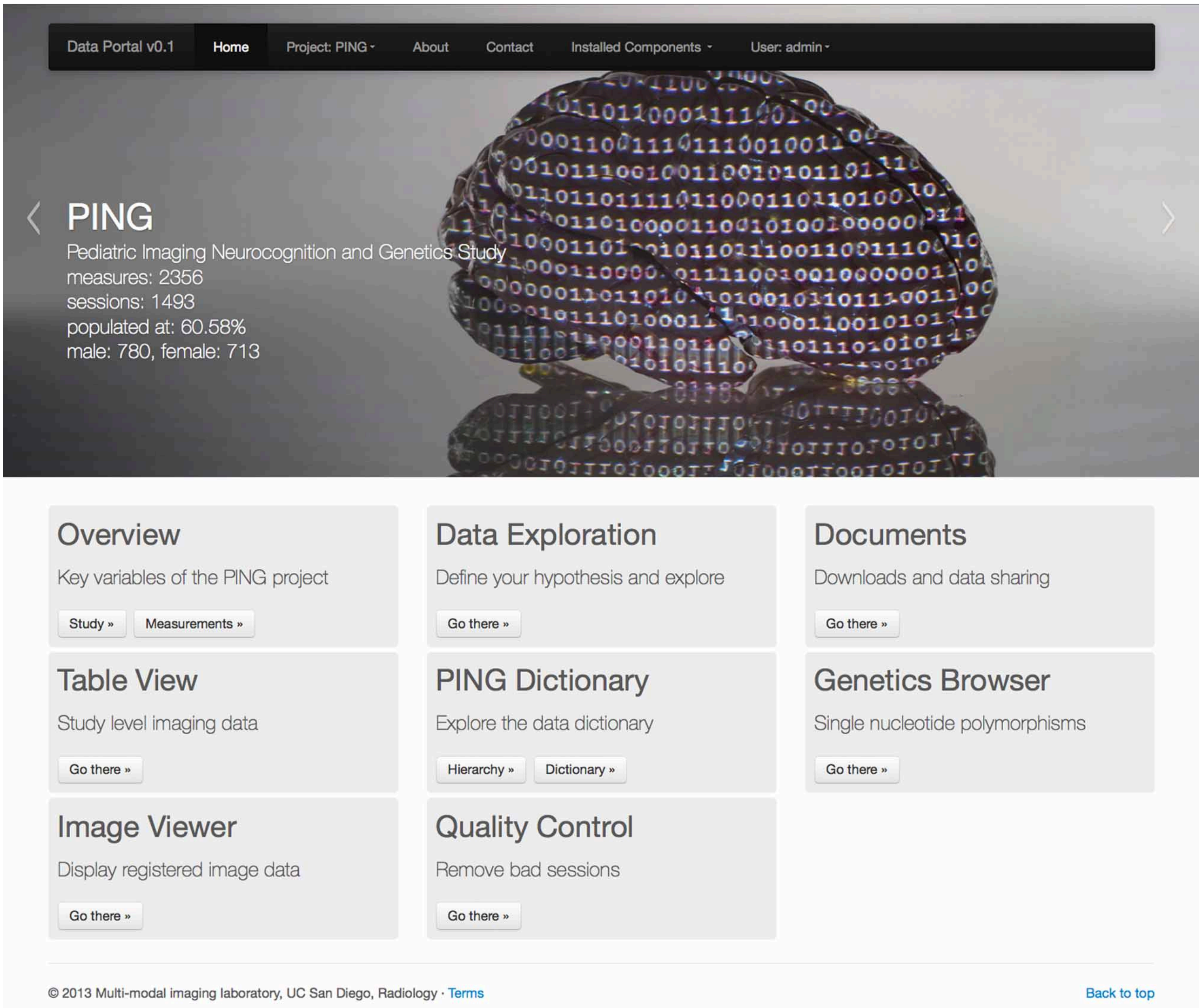
As a secondary workflow the table application helps to identify image data for a known subject identification number. The user can filter the table columns for subject and visit identification number to identify a particular study session.

Applications implement a restricted set of functionalities but provide interfaces that allow them to exchange information with each other. For example, the table application is able to filter data and provides links to the image viewing application. The image viewing application accepts this information and is able to visualize three dimensional reconstructions of multi-modal images.

For this work we refer to collections of subject data as *sessions*. For example, all image data and all the neurocognitive measures obtained during a single visit are collected into a single session that is identified by the subject's identification number and a visit number or date. Typically, the session information is stored in a single row in a data table. *Measures* identify the quantitative or qualitative data obtained for each session and map to columns in this table. Any measure that is not available for a particular session is left empty.

2.2. TECHNOLOGY

The data portal is implemented using a rich client-server, web-based architecture. The web-server delivers data in JavaScript Object Notation (JSON) format together with application code delivered as JavaScript. The clients receive the data, execute the application logic and render the result. Server-side data compression and client-side caching of static data were found to be effective in limiting the resources required on the server (virtual machine with 2GB of main memory and 2 CPU's). The minimum hardware requirements on the client are 1 GHz or faster processor with at least 2 GB RAM and graphics hardware supporting WebGL/OpenGL rendering. The web-interface rendering is implemented using responsive web design and generates appropriate interfaces for workstation computers, laptops,



PING
Pediatric Imaging Neurocognition and Genetics Study
measures: 2356
sessions: 1493
populated at: 60.58%
male: 780, female: 713

Overview
Key variables of the PING project

Data Exploration
Define your hypothesis and explore

Documents
Downloads and data sharing

Table View
Study level imaging data

PING Dictionary
Explore the data dictionary

Genetics Browser
Single nucleotide polymorphisms

Image Viewer
Display registered image data

Quality Control
Remove bad sessions

© 2013 Multi-modal imaging laboratory, UC San Diego, Radiology · Terms

FIGURE 1 | Entry page to the PING data portal reflecting the architecture of the data portal as a collection of workflow driven components. A navigation menu structure and project data summary is displayed in the top half of the page followed by a list of eight application groups. See section 2 for a description of each component.

tablet computers, and smart phones. The application has been successfully tested and is functional on all of these device types.

As a general rule all applications transfer project data from the server to the client machine. The client's browser is responsible for filtering and rendering of the data. All modern browsers have advanced built-in capabilities for data caching, which reduces the dependency of the application on network delay because successive requests can be served from the client's cache. The availability of many JavaScript based libraries for data conversion, analysis, and visualization make it straight forward to adapt novel visualization techniques. An example of a JavaScript library that supports many data visualization tasks is D3 (Bostock et al., 2011). Compute intensive applications such as image analysis cannot be efficiently implemented in JavaScript yet (but see first attempts to improve processing speed by ASM, 2013; Pixastic, 2013). Specialized applications for statistical analysis are also not yet available as a component for web-based architectures. For both of these use cases we integrate server-side processing instead. We will mention in each of the following sections if a server-side implementation was selected.

2.2.1. Server

Server side document storage of structured data is done by text files in either JSON format or in comma-separated-values format (csv) which has been selected as a format of lowest common denominator available at the different data acquisition and processing sites. Whereas more traditional relational databases require an interface to import new data into the data model, our simplified approach stores the original data delivered by each site. As such, updates of the data are synonymous with replacing files and data integrity and versioning are implemented by version control software. Additional information such as user logins, project descriptions and project documentation is stored using JSON notation. This notation is compact enough to be efficient for transport, is supported for automatic parsing on both server and client side and can be viewed and edited as text.

Due to the distributed nature of the PING project, with 10 separate data acquisition sites, it was beneficial to keep a separation of imaging-derived measures (termed *imaging spreadsheet*) and measures related to demographic, genetic and cognitive information (termed *super spreadsheet*). This reduced the dependencies between the research groups handling onsite data acquisition and the group responsible for image data processing as they operated on different schedules. Additionally to the imaging spreadsheet and the super spreadsheet each user of the portal can provide a third, private spreadsheet with supplemental data to integrate derived measures or measures not part of the official dataset (such as site-specific additional measures). Merging of spreadsheets is implemented in the R statistical language (R Core Team, 2013) (freely available software), resulting in an efficient binary, user-specific representation of the study data on the server.

Investigators depend on a stable version of the study data for publication purposes. In order to support reference data sets as well as frequent data updates, the Data Portal provides versioning for uploaded data sets. Users can select the currently active version they wish to work with. As selection of the active version is specific to a browser session users can use this feature to document data differences.

2.2.2. Client

Client applications are built using HTML5 (2013) technology supporting modern application interfaces that run inside standard web-browsers. The client has been successfully tested on Internet Explorer 9.0 (and later), Chrome (version 31 and later), Firefox (version 26 and later), Safari (version 7 and later), and Opera (version 18 and later). The user interface is built using the bootstrap front-end framework (Twitter, 2013) with additional jQuery user interface elements (jQuery, 2013). It provides a consistent look and feel across the different Data Portal applications and supports multiple device types and screen form factors.

2.3. STATISTICAL ANALYSIS

The ability to collate data from multiple sources allows exploration of inter-relationships in the data in a rigorous manner. In order to support online statistical analyses in the Data Portal, we implement an application that combines a web-based interface with server-side statistical processing using R (R Core Team, 2013).

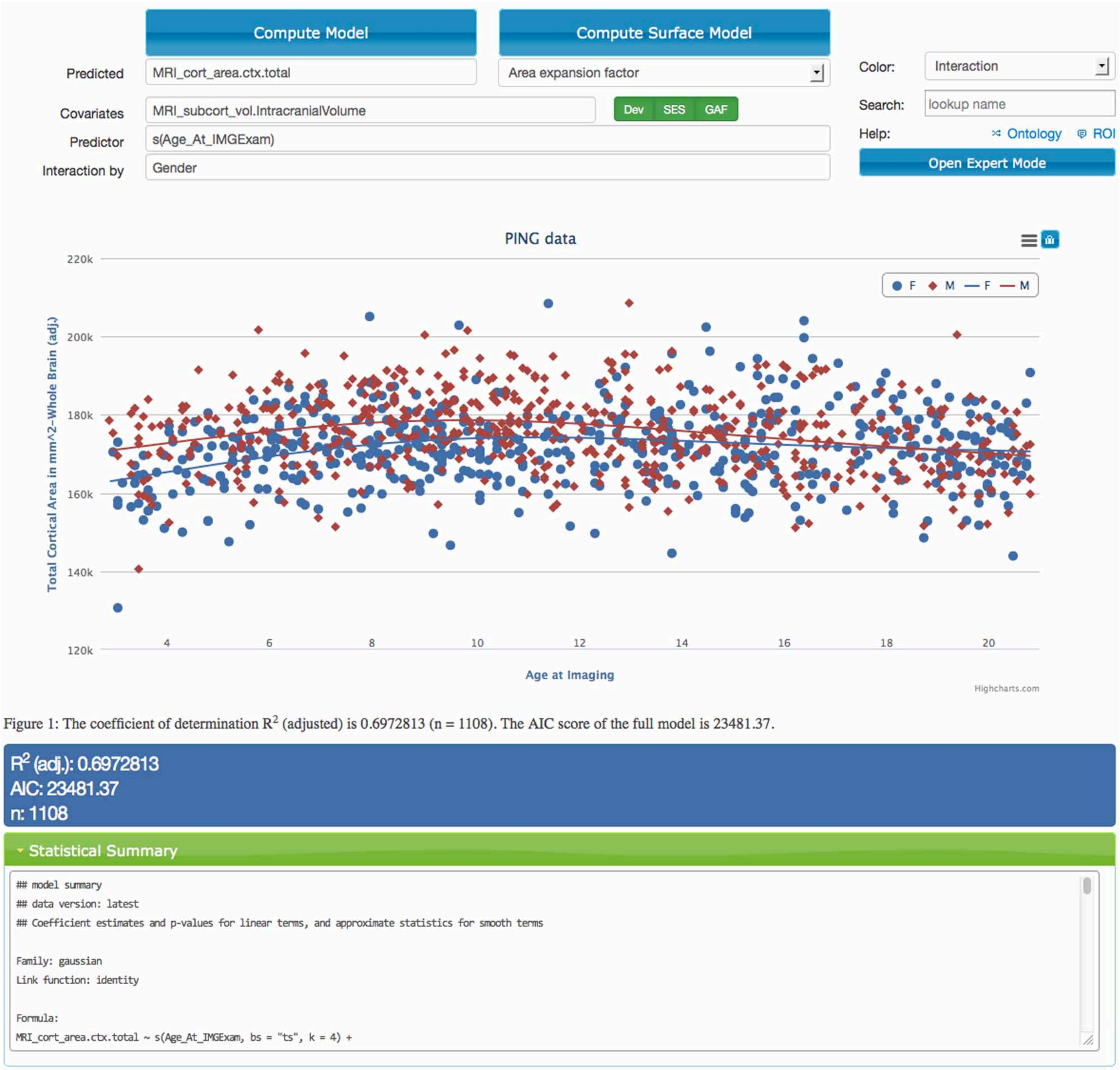
2.3.1. Region of interest based analysis

The application provides input fields organized into a model description mask (see top part of screen capture in **Figure 2**) that allows the user to specify variables of interest (see section 2.5). The application does not require prior knowledge about the syntax used by the R programming language and provides immediate feedback if terms are entered that are not present in the data dictionary. Descriptions for all terms are displayed as tool tips to the user. The model variables include a dependent variable, an independent variable and an arbitrary list of user defined covariates. Additionally the input mask also supports the definition of a separate variable that should interact with the independent variable. In models that are used to describe interactions it is important to include both the main effect of the interaction variable and the interaction term itself which is automatically the case if the interaction field is used.

Additionally, we identified sources of variation known to influence a variety of measures. These *system covariates* include the device serial number of the imaging device, the household income and level of education as socio-economic factors, and genetic ancestry factors derived from gene expression patterns. Users can disable the system covariates, but they are enabled by default (options displayed in green in **Figure 2**). Providing these factors is one way in which domain expert knowledge is implemented in the application. Genetic ancestry factors are encoded as probabilities and are therefore dependent on each other. The system thus automatically removes one of the ancestry groups from the analysis to provide the statistical analysis with the correct degrees of freedom. Utilizing meaningful presets and automatic model extensions in this fashion help to make the statistical analysis application accessible to a wider audience.

Regression analyses are performed on the server using a generalized additive model (GAM) framework with automatic smoothness constrains (Wood, 2013). GAMs include usual linear regression as a special case and are applicable to cross-sectional (single time point) analyses. R reads the project data in binary form, executes the imported model and generates summary measures and model comparison statistics as temporary files. Summary statistics together with model curves and data point coordinates are saved as JSON and transmitted to the client, which is responsible for presenting the data to the user.

Suitable matrix formulations for the computationally intensive parts of the R statistical analysis have been implemented to improve performance of the application. As a further optimization the analysis is restricted to session data for which all model variables are non-missing; sessions with missing values are removed as a first step in the analysis. It is possible in principle to do an initial multiple imputation step for missing data, but this is not currently implemented. The resulting independent and dependent variables are rendered by the client as an interactive scatter plot. Axis labels are inserted using the short description obtained from the data dictionary application, and each data point can be queried using the mouse to display its value and basic demographic information such as gender and age. A link presented to the user for each data point provides a direct connection to the image viewer application that loads relevant session images. Together with the scatter plot, model curves (GAM fits) are displayed in order to provide feedback to the user about the relationship between the dependent and independent variables, including interaction terms. For example, if age is used as an independent variable and gender is used as an interaction term, separate mean curves for males and females are displayed. If the effect of the predictor variable is modeled as a smoothly varying function, the model curves



Predicted: MRI_cort_area.ctx.total

Area expansion factor

Color: Interaction

Search: lookup name

Covariates: MRI_subcort_vol.IntracranialVolume

Predictor: s(Age_At_IMGExam)

Interaction by: Gender

Figure 1: The coefficient of determination R^2 (adjusted) is 0.6972813 (n = 1108). The AIC score of the full model is 23481.37.

R^2 (adj.): 0.6972813
AIC: 23481.37
n: 1108

```
## model summary
## data version: latest
## Coefficient estimates and p-values for linear terms, and approximate statistics for smooth terms

Family: gaussian
Link function: identity

Formula:
MRI_cort_area.ctx.total ~ s(Age_At_IMGExam, bs = "ts", k = 4) +
```

FIGURE 2 | Screen capture of the data exploration application displaying a statistical analysis of the effects of age on the total cortical area for male (red dots and curve) and female (blue dots and curve) children in the PING study. The model corrects for the effects of intra-cranial volume, scanning device, socio-economic factors, and genetic ancestry. Interface components that relate to model specification are shown above the scatter plot. The model is executed on the server using R after selecting the "Compute Model" option. Resulting model curves and residualized data points are plotted together with summary statistics in the middle and lower parts of the web-page. The scatter plot supports an interactive legend, changes in magnification, and data points that link back to imaging data.

might indicate gender specific changes in the predicted variable. The freedom to specify arbitrary variables of interest makes this statistical framework suitable for a wide number of research questions related to age trajectories of brain development. As an example in section 3 we show how to use the PING data portal to analyze the influence of socio-economic factors on imaging measures while correcting for age, gender and genetic factors.

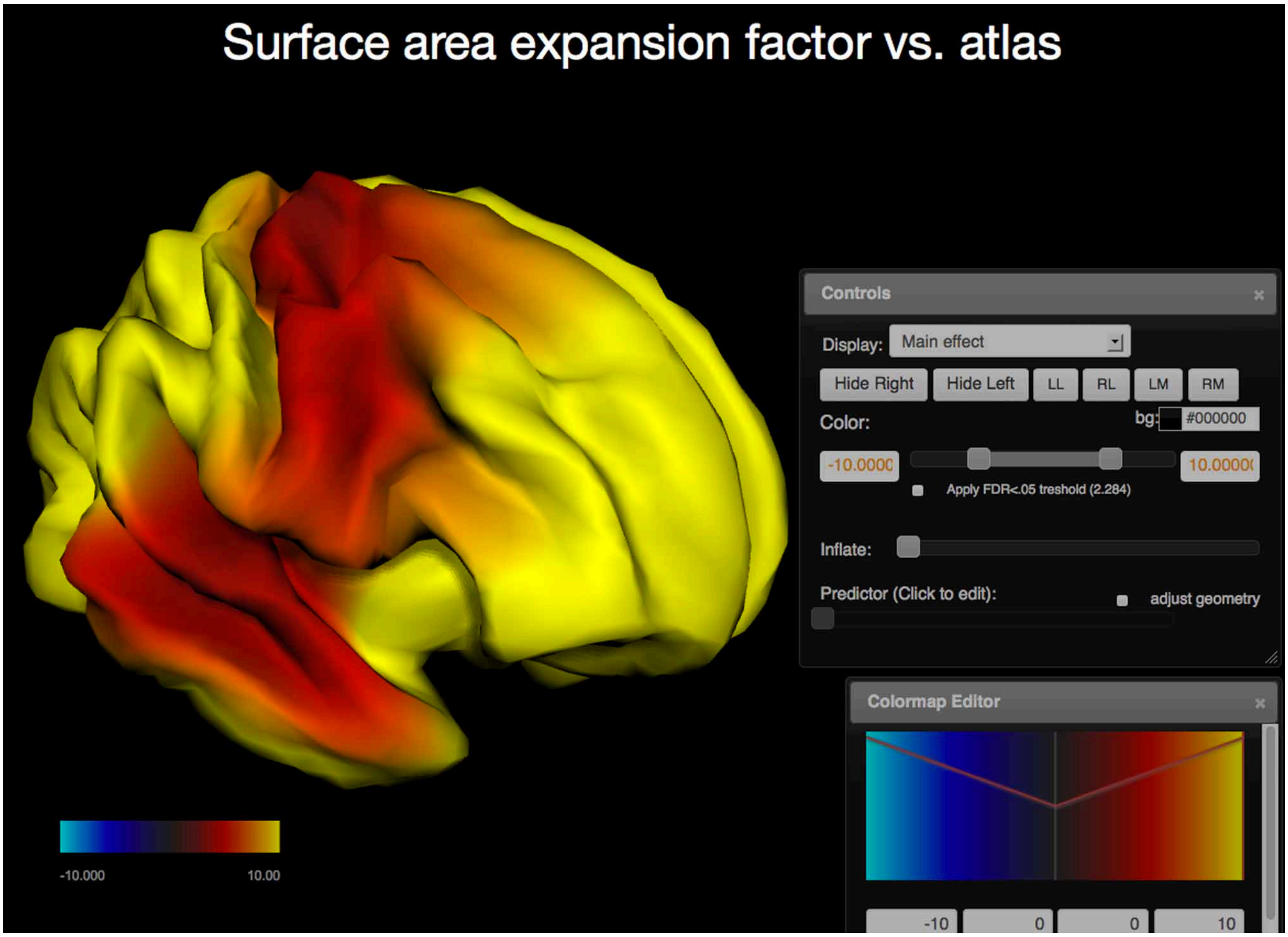
Together with a visual representation using scatter plots and model fits, the application also displays the statistical summary information computed by R (lower part of **Figure 2**). This includes the version number of the data, the generated model specification and the *p*-values for each of the factors. Key model characteristics such as Akaike (1974) and Schwarz's Bayesian information criteria (Schwarz, 1978) are displayed as well and can be used to compare models with different covariates with each other. Both of these model selection procedures help guard against over-fitting by inclusion of too many variables with small effects.

In order to document a particular model users can either export scatter plots in image or spreadsheet format or users can download the data and the R script used for processing. This information can be used to document findings, and users with appropriate knowledge of the statistical models can also alter the script. During development this feature helped in detecting errors created, for example, by inconsistent encoding of measures in the spreadsheets.

2.3.2. Surface based analysis

In addition to region of interest based measures derived from imaging data, the PING study also produced surface based measures for cortical thickness, regional area expansion, and regional volume expansion for each study participant. In this mode the vertex measures are used as dependent variables and the R model is run for each vertex. The resulting surface maps represent regional effect sizes for the (1) user-defined independent variable, (2) the main effect of the interaction variable, if any, (3) its interaction with the independent variable, and (4) estimates for the dependent variables per vertex over the range of the independent variable. Surface maps are written out as JSON and requested by the client. The client renders the surfaces interactively and maps the *p*-values as color (Cabello, 2013; WebGL, 2013); animated maps are used to show the values of the dependent variable over the range of the predictor. The brain geometry is rendered as two independent hemispheres and the user interface provides keyboard shortcuts to allow for the inspection of the inter-hemispheric space (see **Figure 3**).

By default surface maps are rendered using a static surface geometry derived from an atlas brain. The application also provides an option to calculate and display the geometry as a predicted variable. In this mode the surface geometry is deformed


Surface area expansion factor vs. atlas

Controls

Display: Main effect

Color: bg: #000000

-10.0000 10.0000

□ Apply FDR<.05 treshold (2.284)

Inflate:

Predictor (Click to edit): □ adjust geometry

Colormap Editor

-10 0 0 10

FIGURE 3 | Screen capture of the surfer viewer application. Color is used to map the $-\log_{10}(p)$ values of the main effect of age onto each vertex (WebGL cortical surface rendered on the left, same statistical model as in **Figure 2**). The two user interface components displayed are the Colormap Editor (bottom right) which controls a step-wise linear colormap and the "Controls" interface (middle right) that provides a selection of main and interaction effects as well as an option to display the predicted values for each vertex over the range of the predictor (age). Further options include surface re-orientation, background color selection, control of the false discovery rate to correct for effects of multiple comparisons, and an option to adjust the geometry as a predicted variable.

to show the shape trajectories of predicted variables such as age corrected for influences of the selected covariates.

Performing the same statistical analysis for each vertex requires multiple-testing corrections for tests of significance. The application provides for a correction for multiple comparisons using the false-discovery rate (FDR) (Benjamini and Hochberg, 1995). The client uses this information to adjust its color mapping for *p*-value maps using neutral gray tones for regions that are not deemed significant. The user has control over the color mapping and can adjust the colors in the application using step-wise linear transfer functions. Users may also select a point on the surface using the mouse. The name of the corresponding closest region of interest is displayed in that case together with a highlight that shows the outline of the region.

2.4. VIEWING IMAGES

Image data are often acquired in search of biomarkers for diseases. Such biomarkers are derived from anatomical and functional images and are screened by statistical methods for effectiveness in diagnosing disease. In order to get reliable, observer independent measures, automated atlas based image processing pipelines are used (Dale et al., 1999; Hagler et al., 2009). Quality of the generated data depends on appropriate scanning sequences and adherence to scanning protocols. In order to detect protocol violations and other technical anomalies automated and manual quality control of images and derived segmentations are required. This control step is used to identify cases that have to be rejected due to artifacts created for example by subject motion, incorrect scanner settings, or signal dropout. The image viewer application (see **Figure 4**) supports such a quality control workflow by providing a direct link between raw image data and volumes derived after automatic registration and processing. As an example T1 weighted intensity images and color coded cortical and subcortical labels are fused together to allow for a visual inspection of cortical segmentation relative to anatomical scans of T1 image intensity. Scans derived from diffusion weighted imaging are also available as overlays onto anatomical images which supports the inspection of multi-modality registration procedures.

The image viewing application presents a multi-planar reconstruction of volumetric data that is displayed as linked coronal, axial and sagittally oriented images for each modality. A cross-hair tool is used to identify a 3D location in each image stack and the corresponding two orthogonal images closest to this location are loaded from the server and displayed. Scrolling also requests new image tiles from the server. The image storage on the server contains the images for all three orientations registered across several

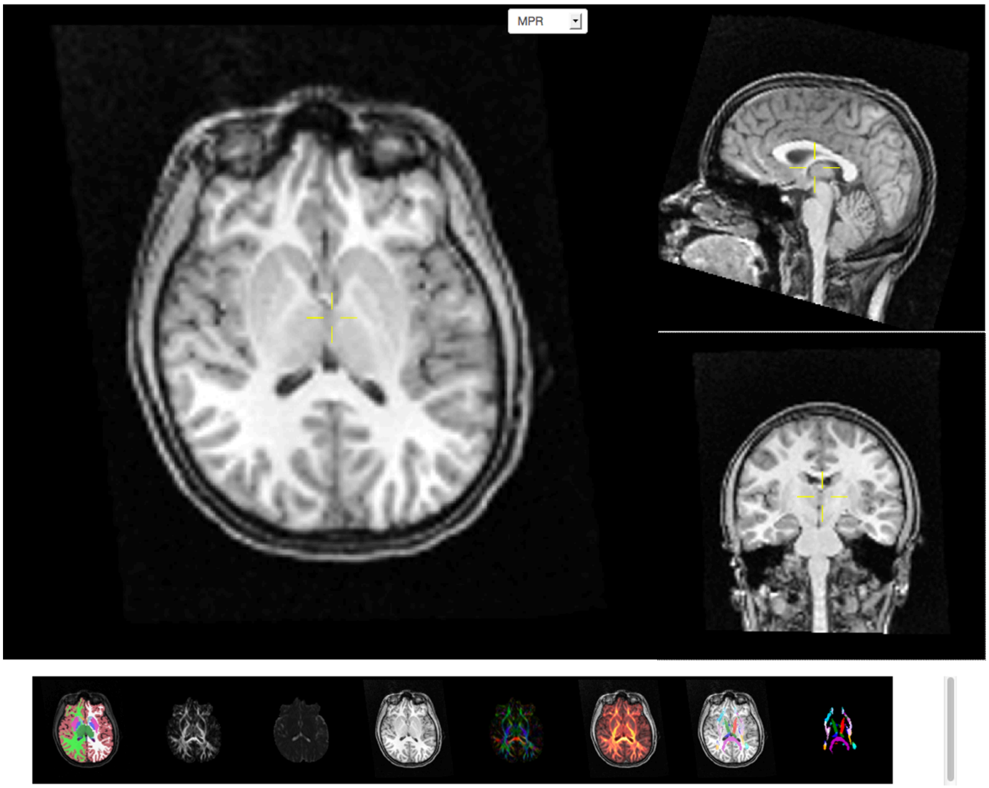


FIGURE 4 | Screen capture of the image viewer application. A multi-planar reconstruction displays axial (top left), sagittal (top right), and coronal (middle right) images linked by a common cross-hair (pale yellow). Below, a row of axial thumbnail images depict available image modalities such as (left to right) fused sub-cortical segmentation with T1-weighted anatomical image, fractional anisotropy (FA), mean diffusivity, T1-weighted anatomical image, color coded directional image stack, fused FA and T1 image stack, fused fiber atlas tract with T1 and fiber atlas tract image stack. All image modalities are registered with each other and selection of a thumbnail image will display the corresponding volumetric information in the multi-planar viewer component above the row of thumbnails. All images support slice browsing using the mouse wheel, brightness, and contrast calibration, and image zoom.

image modalities for each session. The client application displays thumbnail images for each image modality available and switching between modalities allows the user to inspect a specific image location across image modalities. The viewer supports image zoom and pan operations and provides basic adjustments for contrast and brightness. On-demand loading of image tiles and browser caching of already downloaded images allow for interactive performance over a pre-computed image cache containing several millions of images.

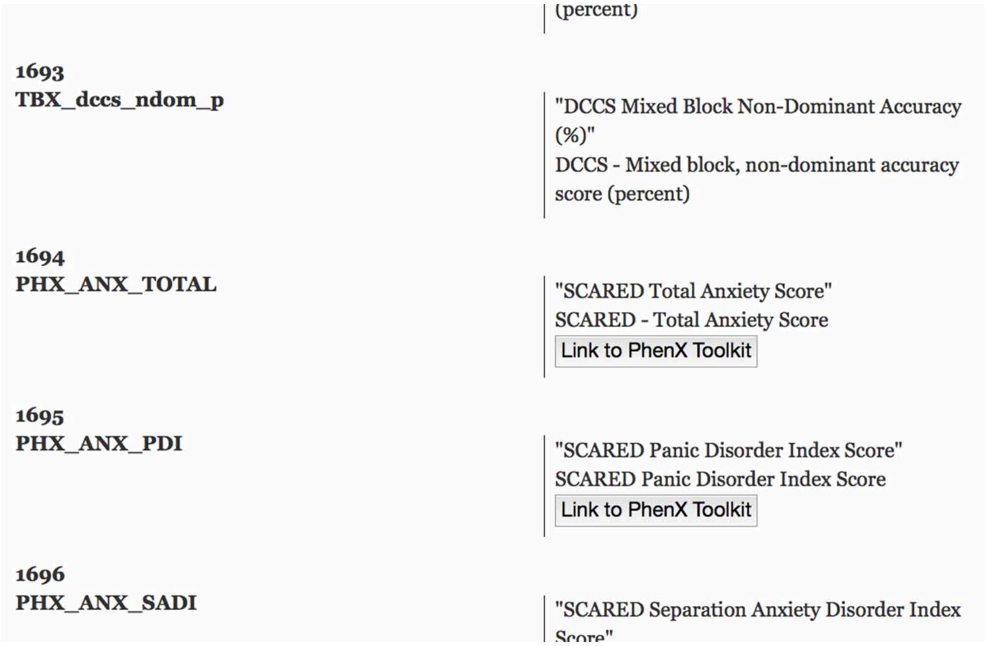
2.5. DATA DICTIONARY

We define a data dictionary as an organization that structures technical terms and their textual descriptions. It is used in this work as the basis for a machine interpretable and processable documentation of the PING related technical terms. The primary use of this resource is to allow researchers not familiar with the PING project to identify measures of interest. The scope of the PING data dictionary is restricted to terms that describe imaging, demographic, self-report, neurocognitive, and genetic measures.

The initial form of the data dictionary is created from the column headers of the data spreadsheets gathered by different units in the research study. Each entry is used as a category (term) with two attributes, a short description which is suitable as an axis label and a textual explanation describing the data encoding in detail. This textual information is made available as a web-service, and data portal application such as the statistical analysis tool utilize this resource.

The data dictionary application provides two visual representations of the data dictionary. All terms are displayed as a list in the *data dictionary view* using HTML5 with embedded RDFa (W3C, 2012, see **Figure 5**). This structured representation allows for data integration and reasoning using external tools. Furthermore, where appropriate, the list displays links to external resources such as the PhenX toolkit (Hamilton et al., 2011). The *structured graph view* facilitates the data exploration of the data dictionary terms. In the PING study more than 2300 measures are available for each study session. Browsing through this collection of terms is supported by imposing a structure that links related terms. The linkage is not exhaustive but merely done in an effort to balance the displayed hierarchy in terms of the number of hierarchy levels and the number of leaf nodes in each category.

We define a term as either a string of characters that is taken from the initial data dictionary or a grouping term that is accompanied by a pattern that maps the grouping term to a subset of all terms. Patterns are implemented as regular expressions utilizing linguistic relationships between terms similar to the work of Ogren (2004). The PING data dictionary lends itself to this analysis as it contains many terms that are derived hierarchically using a small sub-set of root strings. For example, 700 imaging related terms contain the initial string "MRI_" followed by a categorization of the measurement type as either cortial area, thickness, volume, or contrast, followed by an indication for left or right hemisphere and a string characterizing the name of the region of interest. Further examples include self-report measures using the PhenX toolkit (Hamilton et al., 2011) that start with the string "PHX_", genetic ancestry measurements starting with "GAF_", and cognitive measurements obtained using the NIH cognitive toolbox (Weintraub et al., 2013) that start with the letters "TBX_".



(percent)

1693
TBX_dccs_ndom_p

"DCCS Mixed Block Non-Dominant Accuracy (%)"
DCCS - Mixed block, non-dominant accuracy score (percent)

1694
PHX_ANX_TOTAL

"SCARED Total Anxiety Score"
SCARED - Total Anxiety Score

1695
PHX_ANX_PDI

"SCARED Panic Disorder Index Score"
SCARED Panic Disorder Index Score

1696
PHX_ANX_SADI

"SCARED Separation Anxiety Disorder Index Score"

FIGURE 5 | Screen capture of a section of the data dictionary displaying NIH toolbox measures. A sequential number is displayed together with the dictionary term on the left side of the page. On the right side, the corresponding axis label (top) and the available long description (bottom) is listed. Links to external resources such as the PhenX toolkit are embedded into the page. This HTML5 encoded document also contains the RDFa structure information to facilitate knowledge extraction.

Regular expressions are a flexible and efficient way to test large collections of strings. But term clustering using string matching methods is known to be only efficient for small sets of patterns (Tanaka, 1995). Currently about 100 patterns are used to describe the balanced hierarchical structure of the PING project data dictionary. A main requirement for global pattern matching to work is that the string representation of terms used throughout the project is unique. In certain cases we found that terms shared the same name, e.g., "Age" was used to indicate the age at imaging examination and also in a separate group to indicate the age at the neurocognitive examination. In an effort to resolve these cases new terms were introduced to make the entries unique (e.g., "Age_At_IMGExam", "Age_At_NPExam"). This approach to normalization is clearly inefficient and further efforts are needed to include relational types between categories as well as attributes for synonyms and abbreviations. In most cases, if new entries are added to the data dictionary and if those new entries follow already established naming conventions, no change in the pattern set is required to integrate the measurements into the structure. To validate correctness, a coverage check is performed to ensure that (1) the new term is matched by a pattern and (2) appears in the correct place in the hierarchy. If the new term is not correctly matched but conforms to the naming conventions the list of patterns is changed. Changes include the extension of existing patterns to cover the new term using alternations and the introduction of new categories as they are required.

In order to visually represent the derived structure, we use an interactive graph layout (Bostock et al., 2011) which adjusts if elements are added or removed (see **Figure 6**). Only the first two levels of the hierarchy are displayed initially. The user can select a term, the corresponding pattern is executed, and the hierarchy level below the selected term is populated with the matching entries. The layout engine adds the structure using animated unfolding and adjust the spacing between terms. Exploration of the data dictionary structure is therefore interactive and efficient as it only depends on the parts of the hierarchy that the user explores.

2.6. GENETIC INFORMATION

Genotyping using microarrays generates vast amounts of data for each study participant. The size of a typical data vector for each study participant is in the order of 500,000 or more elements. Each of these vectors consists of allele combinations of many SNPs located across the genome. SNP location is specified in terms of number of base pairs from the start of the chromosome. SNP location may overlap with functional regions of the genome that encode genes, pseudo-genes, non-coding RNA, or mRNA sequences. A typical approach to browsing genome-wide data is to search for a particular gene based on findings that relate this gene to a function of interest. SNP values that are captured by the study and which overlap with, or are close to, the gene of interest are selected and used in statistical analyses as independent variables.

The SNP browser provides a user interface that links together gene names and their locations on the chromosome, as well as SNP locations and allele combinations for each study participant (see **Figure 7**). We found that network speed and modern browser technology easily keep up with the transmission of data generated by larger studies. Data can be stored on the client computer in memory but browser-based applications are limited in terms of their ability to simultaneously display graphical representations often involving thousands of objects. We solve this problem by using client-based logic that prevents content from being rendered that is not currently visible in the browser window. User interactions like scrolling are used as a signal to add content. As an example, the SNP browser application appears to list initially all 500,000 SNP's in a single table as no search term is specified at the start of the application. As the user scrolls down the page, data are dynamically added to the bottom using unobtrusive pagination. This approach limits the number of items rendered in the browser and adjusts naturally with differences in screen size and resolution of client machines.

The user can search for a specific gene using its name or a suitable regular expression that is matched against all gene names. The client application filters the global set of names and populates the chromosome and the base pair range fields in the interface.

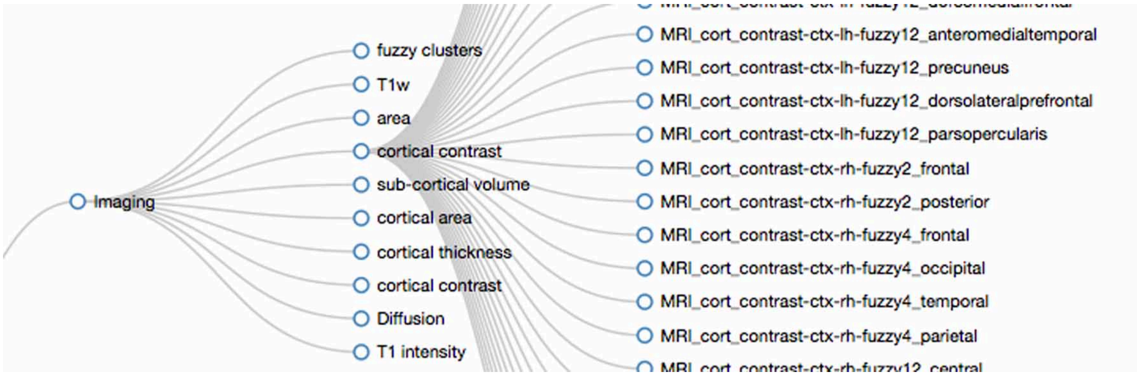
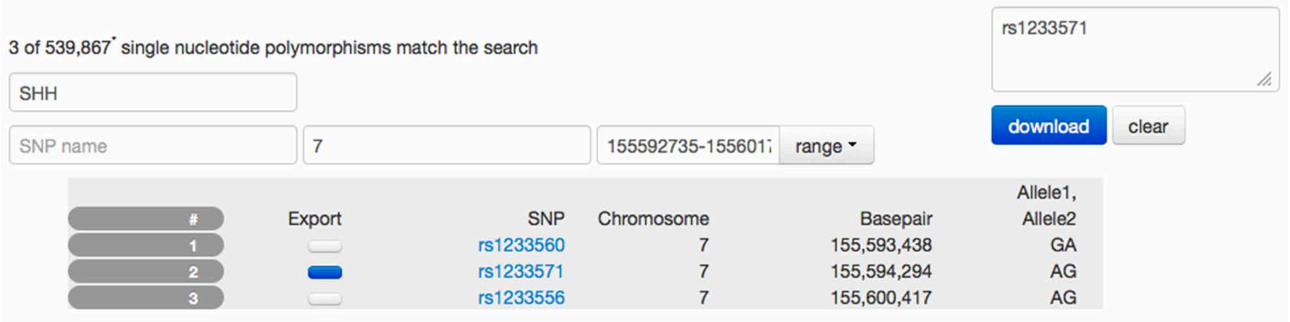

Imaging
fuzzy clusters
T1w
area
cortical contrast
sub-cortical volume
cortical area
cortical thickness
cortical contrast
Diffusion
T1 intensity
MRI_cort_contrast-ctx-lh-fuzzy12_anteromedialtemporal
MRI_cort_contrast-ctx-lh-fuzzy12_precuneus
MRI_cort_contrast-ctx-lh-fuzzy12_dorsolateralprefrontal
MRI_cort_contrast-ctx-lh-fuzzy12_parsopercularis
MRI_cort_contrast-ctx-rh-fuzzy2_frontal
MRI_cort_contrast-ctx-rh-fuzzy2_posterior
MRI_cort_contrast-ctx-rh-fuzzy4_frontal
MRI_cort_contrast-ctx-rh-fuzzy4_occipital
MRI_cort_contrast-ctx-rh-fuzzy4_temporal
MRI_cort_contrast-ctx-rh-fuzzy4_parietal
MRI_cort_contrast-ctx-rh-fuzzy12_central

FIGURE 6 | Screen capture displaying parts of the hierarchical structure of the PING data dictionary. The branches for "Imaging" and "cortical contrast" have been opened by the viewer. The regular expression used to create the displayed hierarchy level for "Imaging" is "/(H_area | H_thickness | H_contrast | H_volume | H_intensity | Diffusion | H_Fuzzy)/". The entry "cortical contrast" (H_contrast) is implemented by the pattern "/(^MRI_cort_contrast)/". In PING this maps to all MRI related cortical contrast measures in the data dictionary (subset displayed on the right).



3 of 539,867* single nucleotide polymorphisms match the search

SHH

SNP name | 7 | 155592735-155601; range

rs1233571

#	Export	SNP	Chromosome	Basepair	Allele1, Allele2
1	☐	rs1233560	7	155,593,438	GA
2	☑	rs1233571	7	155,594,294	AG
3	☐	rs1233556	7	155,600,417	AG

FIGURE 7 | Screen capture of the SNP browser application used to explore and extract genetic information available for the PING study. A search mask is used to specify a gene (SSH, sonic hedgehog). Utilizing a database with 80,000 entries, the SNP browser obtains the available chromosome number (7) and the basepair location (155,592,735–155,601,766) for this gene. The table is filled with SNP entries that fall in the range of the basepair location. In this example, three SNP entries are available. The user has selected SNP number 2 indicated by the dark blue checkbox and the corresponding SNP name has been copied to the list of SNP names for download. Selecting the download option would provide the user with a spreadsheet of the alleles for this SNP for all PING subjects.

This search is performed on the client computer using cached data and does not require resources on the server or even a connection to the server. The resulting table displays the subset of SNP locations that fall into the base pair range indicated by the gene. Each of the SNPs displayed is linked to the NCBI (2013) database for further information.

The SNP browser provides access to the study specific SNP data so users can select candidate SNPs for further analysis. The SNP names are collected as a editable list in the interface and, upon request, the list of SNP alleles for each subject id is generated and presented to the user as a csv spreadsheet for download. Due to the size of the SNP database (approximately 2 gigabytes of binary data) a server-side implementation using the *PLINK* software (Purcell et al., 2007) is used to create each spreadsheet. PLINK provides an efficient binary storage for large samples of SNP data which reduces storage requirements and, more importantly, provides a fast read and access to SNP information. Updates of SNP data are supplied as PLINK files, which are copied directly to the server. No further processing is required to integrate the information into the Data Portal. Processing time on the server is similar to the time required to download the resulting spreadsheet.

The SNP browser exports data that are suitable for upload into the statistical analysis application of the Data Portal. Together with genetic ancestry information already available in the statistical analysis application, this setup provides a flexible solution allowing genetic information to be linked to other data domains.

2.7. QUALITY CONTROL

Several applications are suitable to detect outliers in the data. For example, the image viewing application shows region of interest (brain labels) merged with anatomical information. Errors in the segmentation are easily identifiable on these images. Outliers on the population level are apparent in the scatter plots of the data exploration application. Each data point in scatter plots links to the corresponding imaging data. Exploring these data can increase confidence that observed variation is due to true variation between study participants and is not caused by differences in image quality or noise levels.

A dedicated application is used to capture the current status of quality control. Sessions with known problems are indicated and textual annotations are used to explain choices of inclusion or exclusion of data from analysis. Access to the quality control application is limited to trained personnel using a role-based access control system. If a data point is marked as "bad," it is excluded from further analysis by forcing a re-generation of the data cache available to each user. Results of the process are immediately accessible to every user of the portal.

2.8. SECURITY AND PRIVACY CONSIDERATIONS

Removal of patient identifying data is performed during data acquisition as stipulated by the responsible institutional review boards. Only limited information is provided to researchers requesting access. However, the combination of many sources of information such as genetic, demographic, and imaging data poses unique challenges for data privacy.

For example, genomic data have the potential to link people across studies (Homer et al., 2008). These data could therefore be used to re-identify study participants or their relatives. In order to prevent such activities and to ensure the privacy of study participants, access to the Data Portal is secured. Each user of the Data Portal is required to agree to a data use policy that forbids the use of study data for the purpose of de-identification. As a further precaution the SNP viewer implements limits on the number of SNP values that a user can download for a given project. The download of all SNP values is not supported by the data portal but may be provided using dedicated data sharing sites for genetic information like dbGAP (Mailman et al., 2007).

3. RESULTS

The large number of variables available in the PING study provide a rich resource for data exploration of patterns of brain development. As an example we show how to explore a measure of cortical surface area over age. Area of the heavily folded human cortex correlates under some circumstances with the number of neurons available for processing and has been implicated as a variable of interest for describing the developing human brain.

Identifying variables of interest: The first step is to identify data dictionary entries that refer to the variables of interest. Using the data dictionary application a measure of the total cortical thickness called "MRI_cort_area.ctx.total" can be found in the sub-tree *imaging*. Additionally, the entry is also listed in the section labeled *summary measures*. If parts of the variable name are known the data exploration application input fields can be used as search fields that display the matching content in drop-down lists. Entering the search strings "area" or "total" would list the measure together with a textual description as a tool tip.

Initially we start with the simplest model by disabling the system defined covariates that represent imaging device identity, socio-economic factors and genetic ancestry factors. After specifying the independent and dependent variables in the data exploration application as total cortical area and age at image examination the model is executed and produces a linear functional relationship between cortical surface area and age. This initial model shows a large spread of the scatter points which indicates a poor explanatory power of the model which is confirmed by a low value of variance explained (0.076%) as listed in the statistical summary section. Only a small part of the relationship between brain surface area and age can be explained by our initial model.

Model comparison: Replacing the linear function of age "Age_At_IMGExam" with a smoothly varying function "s(Age_At_IMGExam)" we can improve the fit. The new model captures an initial increase in total cortical area followed by decreasing cortical area over age (variance explained 7.5%).

Adding back the sequence of system covariates for imaging device, socio-economic factors (household income, highest level of parental education) and genetic ancestry the models can capture successively more of the variance (13, 17, 20%). It is well known that adding variables to a model will tend to increase its explanatory power which at some point will lead to poor generalization as accidental features of the data are captured. Also, as new measures are added a subset of subjects will need to be removed from the analysis if measurements are not available for them. In order to be able to detect over-fitting the data exploration application displays the adjusted coefficient of determination $\bar{R}^2$ which incorporates a correction for the number of variables included into the model. Increasing values over successive runs of the model confirm that our model variables help to explained the observed variance without introducing over-fitting. This is also confirmed by the displayed Akaike Information Criterion (AIC, Akaike, 1974).

Whereas the system covariates have been identified as sensible choices for model testing it is up to the investigator to identify further variables. A potential source of variation not captured by our current model is gender differences. Also, cortical area will likely scale with head size so our measure for total surface area includes effects that can be attributed to varying head sizes. Identifying measures for gender and intra-cranial volume and including them into the model increases variance explained to 70%.

Significance analysis: The significance of each model variable is listed in the statistical summary section of the data exploration application. The socio-economic factors for example show that effects of household income are significant ($p < 0.01$) whereas the level of the highest parental education is not.

Analysis of interaction: Adding gender as a covariate explained a large part of the variance observed. This could indicate that there is substantial difference in the developmental patterns of male and female children. In order to investigate these differences the data portal can calculate interaction effects with age and display separate model curves for each gender. Moving gender from the covariate text field to the field labeled *interaction* both the main effect of gender and the interaction effect of gender and age are added to the model. We observe a highly significant interaction effect of age by gender ($p < 0.001$). The total cortical area over age curves generated by executing this model suggest that the developmental trajectories differ for boys and girls. Cortical area appears to peak slightly earlier and decline somewhat more rapidly in boys than in girls.

Adding data sources: Using the rich literature of genes implicated in development and disease we can use the portal to try to replicate findings using the PING data. The STON2 gene has been implicated as being correlated with regional surface area in a model of schizophrenia Xiang et al. (2013). The SNP Browser application of the Data Portal provides access to selected SNPs that are located in genes of interest. A search for "STON2" reveals three SNPs that are located on chromosome 14 and intersect with STON2. Upon request the application generates a csv formatted spreadsheet with three columns of SNP alleles for each PING subject for download. The spreadsheet format is such that it can be uploaded into the data exploration application as a user defined spreadsheet. Adding the SNP names as new covariates we can create a model that tests for significance of SNP allele combinations related to STON2 on total cortical area. Running the extended model reveals no significant effect of STON2 SNP alleles on total cortical area in the PING study. Other external sources of subject specific information can be integrated into the data exploration portal in a similar manner.

Surface based analysis: Cortical area measures are available in the PING study as atlas-based regions of interest measures (Desikan et al., 2006). Additionally, area measures are available for each point on the cortical surface. These measures are calculated as factor values of local area expansion required to map points in the individual brain onto points in an atlas brain. Regions of the brain that need to expand if mapped to the atlas can therefore be distinguished from regions that would need to contract.

The surface based analysis uses the same model description and statistical tools as the region based analysis. Our current model description can therefore also be used to calculate developmental effects of regional surface area. The surface viewer application displays the calculated p-values as surface maps of $-\log_{10}$ scaled p-values (see **Figure 3**). The scaling is used to map high significance (low p-values) to large false-color encoded values indicating regions which are significant at a level of $p < 0.05$. Running the model simultaneously for each point on the surface may cause some regions to reach significance due to chance alone (multiple testing). To counteract this artificial inflation of significance the surface viewer application performs re-scaling of the *p*-value surface maps using false-discovery rate (FDR) calculations. Similar features are available for surface-based estimates of cortical thickness and volume.

Inspection of the calculated point-wise surface area expansion factor due to age shows a general increase for all points relative to the given atlas. This is explained by the choice of the atlas which represents an adult brain. A measure independent of the choice of the atlas is available in the surface viewer application as the instantaneous rate of change. This measure highlights cortical areas that show increased (red to yellow) or decreased (blue) developmental change at a particular age (dependent variable of the model). Using this option the surface viewer animates the complex pattern of developmental change over age for both hemispheres and provides presets for the visual inspection of both hemispheres and the inter-hemispheric spaces.

Export: The graphic system used to display the model and scatter plot adjusts to the display size of the device used to view the page. Model curves and point sets can be switched on and off independently using controls embedded in the legend of the figure. Each scatter plot point can be investigated and reveals session information such as the anonymized subject identification number, the gender and links to the image viewer application.

The scatter plot and model functions can be exported as vector graphics (pdf format) or, as quantitative data (csv format) for further statistical analysis. The data exploration application also provides a download package that contains the source code of the statistical analysis and the study data in R format. This package can be used for both documentation and replication of the implemented statistical methods.

Each map displayed by the Surface Viewer can be exported as a high quality graphic (png image format) with transparency used to encode background pixel. The graphics can easily be assembled into publication ready representations of key developmental figures (see **Figure 8** for image collage).

Efficient implementation of statistical methods on the server reduces the response time for a full statistical analysis to approximately 5–10 s. The combination of automatic model generation for the R programming language, server side execution, export of result data and the integration of visualization and data exploration provides a low barrier of entry for people with limited technical expertise. The features implemented in the data portal therefore extend the usability of study data to a larger audience.

4. DISCUSSION

Traditional approaches for data management have used database management system to store all information related to a project. In this setup data import and export algorithms become the tools to map domain specific data to structures suitable for database storage and retrieval. The specific choice for the database layout, such as the number of tables and the values, keys and indexes that are stored in a relational database is expected to be stable over time. This requires projects to make decisions early on, often using insufficient information. Using relational databases therefore can be costly if changes in the database layout are required. Often application logic for import and export of data is used instead to adjust to changing requirements.

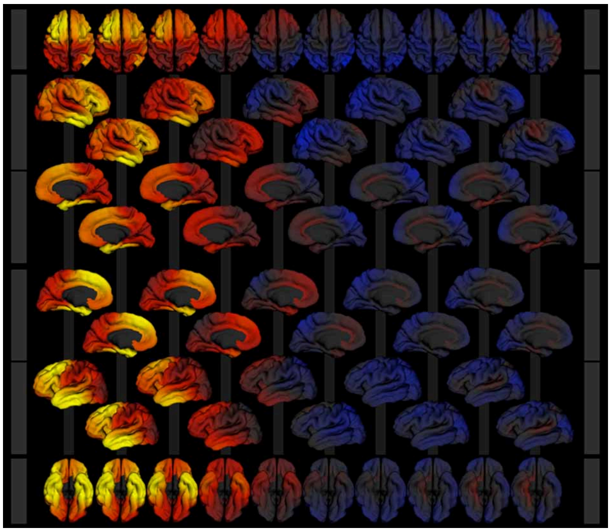


FIGURE 8 | Image collage of surface models exported from the surface viewer application for the model described in section 3. Cortical area expansion factor is mapped as color (red—expansion, blue—contraction) over age (3–21 years, left to right) given the model described in section 3. Rows show superior (1), right lateral (2 and 3), medial view of the right hemisphere (4 and 5), medial view of the left hemisphere (6 and 7), left lateral (8 and 9), and inferior (10) views of the 3d surface model.

The data portal architecture improves on this approach by favoring file formats that are linked to data processing and visualization. These data structures augment the database management infrastructure as a primary source for algorithmic processing. For example, binary representations of large datasets such as genomic data combine compact representation and guarantee fast access while minimizing system resources. Our approach of keeping established data formats such as PLINK's binary format and R's RData format for storage on the server has also simplified the incremental update of our system to new versions of the data because only a small number of files have to be replaced using a very simple procedure.

The PLINK and R software applications are publicly available and provide efficient read access for compact data caches minimizing server requirements on memory and speed. Furthermore, data are stored in a way that is best suited to the specific application that implements the data analysis. Instead of data warehousing with complex implementations of data access, domain specific languages provide an integration for suitable analysis algorithms. In this framework the domain languages are responsible for translation of processing results into formats suitable for transfer and decoding on the client side (JSON, csv).

One of the more challenging aspects of the data portal development has been the efficient transfer of large assemblies of image data. The application preserves bandwidth by downloading only images that are displayed and images that are immediately adjacent. One way to increase viewing performance further is to combine images into mosaics. This would result in a lower number of file transfers with larger files which is more efficient in the setting of web-based applications.

The statistical processing is optimized for cross-sectional studies. Longitudinal analysis requires the use of an extended statistical modeling framework, such as generalized additive mixed models. Currently the data portal is able to detect longitudinal data and data points in the scatter plot that belong to the same participant are marked. A warning is presented to the user that informs him or her about the restrictions imposed by the cross-sectional analysis stream.

The data portal combines the elements of study participant specific exploration of key developmental features across behavioral, imaging and genetics domains with capabilities to formulate and test study level hypotheses and estimate population parameters.

ACKNOWLEDGMENTS

The authors would like to thank the users of the PING data portal for providing the feedback required to develop the data portal, Trish Whetzel for providing feedback on the data dictionary section in this paper and Daniel Marcus for his encouragements to publish this work. The brain image used in **Figure 1** has been graciously provided by J. Annese and C. Sheh from the UCSD Brain Observatory.

FUNDING

RC2 DA029475; R01 HD061414/HD/NICHD NIH HHS/United States.

REFERENCES

1000 Genomes. (2012). *An Integrated Map of Genetic Variation From 1,092 Human Genomes*. doi: 10.1038/nature11632

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Automat. Contr.* 19, 716–723. doi: 10.1109/TAC.1974.1100705

ASM. (2013). *ASM*. Availble online at: http://asmjs.org/, Last viewed July 2013.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B* 57, 289–300.

Biffi, A., Anderson, C. D., Desikan, R. S., Sabuncu, M., Cortellini, L., Schmansky, N., et al. (2010). Genetic variation and neuroimaging measures in alzheimer disease. *Arch. Neurol.* 67, 677–685. doi: 10.1001/archneurol.2010.108

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3: data-driven documents. *IEEE Trans. Vis. Comp. Graph.* 17, 2301–2309. doi: 10.1109/TVCG.2011.185

Buccigrossi, R., Ellisman, M., Grethe, J., Haselgrove, C., Kennedy, D. N., Martone, M., et al. (2008). The neuroimaging informatics tools and resources clearinghouse (NITRC). *AMIA Annu. Symp. Proc.* 6, 1000.

Cabello, R. (2013). *Three.js (r59)*. JavaScript 3D library. Available online at: http://mrdoob.github.com/three.js/

JMP. (1989–2007). *Version 7*. SAS Institute Inc., Cary, NC, 1989–2007.

Dale, A., Fischl, B., and Sereno, M. I. (1999). Cortical surface-based analysis: i. segmentation and surface reconstruction. *Neuroimage* 9, 179–194. doi: 10.1006/nimg.1998.0395

DCM4CHEE (2013). *DCM4CHEE*. Available online at: http://www.dcm4che.org/, Last viewed July 2013.

Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., et al. (2006). An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage* 31, 968 – 980. doi: 10.1016/j.neuroimage.2006.01.021

Fjell, A. M., Walhovd, K. B., Brown, T. T., Kuperman, J. M., Chung, Y., Hagler, D. J., et al. (2012). Multimodal imaging of the self-regulating developing brain. *Proc. Natl. Acad. Sci. U.S.A.* 109, 19620–19625. doi: 10.1073/pnas.1208243109

Hagler, D. J., Ahmadi, M. E., Kuperman, J., Holland, D., McDonald, C. R., Halgren, E., et al. (2009). Automated white-matter tractography using a probabilistic diffusion tensor atlas: application to temporal lobe epilepsy. *Hum. Brain Mapp.* 30, 1535–1547. doi: 10.1002/hbm.20619

Hamilton, C. M., Strader, L. C., Pratt, J. G., Maiese, D., Hendershot, T., Kwok, R. K., et al. (2011). The PhenX Toolkit: get the most from your measures. *Am. J. Epidemiol.* 174, 253–260. doi: 10.1093/aje/kwr193

Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., and Conde, J. G. (2009). Research electronic data capture (REDCap) - a metadata-driven methodology and workflow process for providing translational research informatics support. *J. Biomed. Inform.* 42, 377–381. doi: 10.1016/j.jbi.2008.08.010

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., et al. (2008). Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet.* 4:e1000167. doi: 10.1371/journal.pgen.1000167

HTML5. (2013). *HTML5*. Available online at: http://dev.w3.org/html5/, Last viewed July 2013.

Hurst, L., Stafford, M., Cooper, R., Hardy, R., Richards, M., and Kuh, D. (2013). Lifetime socioeconomic inequalities in physical and cognitive aging. *Am. J. Public Health* 103, 1641–1648. doi: 10.2105/AJPH.2013.301240

jQuery. (2013). *jQuery*. Available online at: http://jquery.com, Last viewed July 2013.

Mailman, M. D., Feolo, M., Jin, Y., Kimura, M., Tryka, K., Bagoutdinov, R., et al. (2007). The NCBI dbGaP database of genotypes and phenotypes. *Nat. Genet.* 39, 1181–1186. doi: 10.1038/ng1007-1181

Marcus, D. S., Olsen, T., Ramaratnam, M., and Buckner, R. L. (2007). The extensible neuroimaging archive toolkit (XNAT): an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34.

Monzalvo, K., Fluss, J., Billard, C., Dehaene, S., and Dehaene-Lambertz, G. (2012). Cortical networks for vision and language in dyslexic and normal children of variable socio-economic status. *Neuroimage* 61, 258–274. doi: 10.1016/j.neuroimage.2012.02.035

NCBI. (2013). *National Center for Biotechnology Innovation*. Available online at: http://www.ncbi.nlm.nih.gov, Last visited July 2013.

Ogren, P. V. (2004). "The compositional structure of gene ontology terms," in *Biocomputing-Proceedings of the 2004 Pacific Symposium*, (Singapore), 214–225.

OpenClinica LLC, and collaborators (2013). *OpenClinica*. Available online at: http://www.openclinica.com, Last viewed July 2013.

Pixastic. (2013). *Pixastic*. Available online at: http://www.pixastic.com/lib/, Last viewed July 2013.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A. R., Bender, D., et al. (2007). PLINK: a toolset for whole-genome association and population-based linkage analysis. *Am. Jo. Hum. Genet.* 81, 559–575. doi: 10.1086/519795

R Core Team. (2013). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.

Rosset, A., Spadola, L., and Ratib, O. (2004). Osirix: an open-source software for navigating in multidimensional DICOM images. *J. Digit. Imaging* 17, 205–216. doi: 10.1007/s10278-004-1014-6

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Stat.* 6, 461–464. doi: 10.1214/aos/1176344136

Tanaka, E. (1995). Theoretical aspects of syntactic pattern recognition. *Pattern Recog.* 28, 1053–1061. doi: 10.1016/0031-3203(94)00182-L

Tukey, W. J. (1980). We need both exploratory and confirmatory. *Am. Statist.* 34, 23–25. doi: 10.2307/2682991

Twitter. (2013). *Bootstrap*. Available online at: http://getbootstrap.com/, Last viewed July 2013.

W3C. (2012). *Rdfa Lite 1.1*. Available online at: http://www.w3.org/TR/rdfa-lite/, Last visited Feb 2014.

Wang, F., Thiel, F., Furrer, D., Vergarra-Niedermayr, C., Qin, C., Hackenberg, G., et al. (2008). An adaptable XML based approach for scientific data management and integration. *Proc. SPIE* 6919, 69190K-1–69190K-10 doi: 10.1117/12.773154

Weasis. (2013). *Weasis*. Available online at: http://www.dcm4che.org/confluence/display/WEA/Home, Last viewed July 2013.

WebGL (2013). *WebGL*. Available online at: http://www.khronos.org/registry/webgl/specs/latest/, Last viewed July 2013.

Wechsler, D. (2004). *Wechsler Scale for Intelligence*. 4th Edn. London: Pearson Assessment.

Weintraub, S., Dikmen, S. S., Heaton, R. K., Tulsky, D. S., Zelazo, P. D., Bauer, P. J., et al. (2013). Cognition assessment using the NIH toolbox. *Neurology* 80(11 Suppl. 3), S54–S64. doi: 10.1212/WNL.0b013e3182872ded

Wood, S. (2013). *Mixed GAM Computation Vehicle With GCV/AIC/REML Smoothness Estimation*. Available online at: http://cran.r-project.org/web/packages/mgcv/index.html, Last viewed July 2013.

Xiang, B., Wu, J. Y., Wang, Q., Li, M. L., Jiang, L. J., Deng, W., et al. (2013). Cortical surface area correlates with STON2 gene Ser307Pro polymorphism in first-episode treatment-naive patients with schizophrenia. *PLoS ONE* 8:e64090. doi: 10.1371/journal.pone.0064090

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 10 December 2013; accepted: 25 February 2014; published online: 26 March 2014.

Citation: Bartsch H, Thompson WK, Jernigan TL and Dale AM (2014) A web-portal for interactive data exploration, visualization, and hypothesis testing. Front. Neuroinform. ***8****:25. doi: 10.3389/fninf.2014.00025*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 27 March 2014
doi: 10.3389/fninf.2014.00030

The multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) high performance computing infrastructure: applications in neuroscience and neuroinformatics research

Wojtek J. Goscinski[1]*, Paul McIntosh[1], Ulrich Felzmann[2], Anton Maksimenko[2], Christopher J. Hall[2], Timur Gureyev[3], Darren Thompson[3], Andrew Janke[4], Graham Galloway[4], Neil E. B. Killeen[5], Parnesh Raniga[6,7], Owen Kaluza[1,6], Amanda Ng[1,6,8], Govinda Poudel[6], David G. Barnes[1,6,8], Toan Nguyen[6], Paul Bonnington[1] and Gary F. Egan[6]

[1] *Monash eResearch Centre, Monash University, Clayton, VIC, Australia*
[2] *Australian Synchrotron, Clayton, VIC, Australia*
[3] *CSIRO, Clayton, VIC, Australia*
[4] *Centre for Advanced Imaging, University of Queensland, St Lucia, QLD, Australia*
[5] *The University of Melbourne, Melbourne, VIC, Australia*
[6] *Monash Biomedical Imaging, Monash University, Clayton, VIC, Australia*
[7] *CSIRO Preventative Health Flagship, CSIRO Computational Informatics, The Australian e-Health Research Centre, Herston, QLD, Australia*
[8] *Life Sciences Computation Centre, VLSCI, Parkville, VIC, Australia*

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Antonio C. Roque, Universidade de São Paulo, Brazil
B. Nolan Nichols, University of Washington, USA

****Correspondence:***
Wojtek J. Goscinski, Monash eResearch Centre, Monash University, Building 75, Clayton, VIC 3168, Australia
e-mail: wojtek.goscinski@monash.edu

The Multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) is a national imaging and visualization facility established by Monash University, the Australian Synchrotron, the Commonwealth Scientific Industrial Research Organization (CSIRO), and the Victorian Partnership for Advanced Computing (VPAC), with funding from the National Computational Infrastructure and the Victorian Government. The MASSIVE facility provides hardware, software, and expertise to drive research in the biomedical sciences, particularly advanced brain imaging research using synchrotron x-ray and infrared imaging, functional and structural magnetic resonance imaging (MRI), x-ray computer tomography (CT), electron microscopy and optical microscopy. The development of MASSIVE has been based on best practice in system integration methodologies, frameworks, and architectures. The facility has: (i) integrated multiple different neuroimaging analysis software components, (ii) enabled cross-platform and cross-modality integration of neuroinformatics tools, and (iii) brought together neuroimaging databases and analysis workflows. MASSIVE is now operational as a nationally distributed and integrated facility for neuroinfomatics and brain imaging research.

Keywords: neuroinformatics infrastructure, high performance computing, instrument integration, CT reconstruction, cloud computing, Huntington's disease, Quantitative susceptibility mapping, digital atlasing

INTRODUCTION

The "21st century microscope" will not be a single instrument; rather it will be an orchestration of specialized imaging technologies, data storage facilities, and specialized data processing engines. Moreover, scientists increasingly require access to a wide range of imaging instruments, across multiple modalities and multiple scales, to characterize a scientific sample or perform an experiment. The Multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE—www.massive.org.au) is a high performance computing facility that is specialized for computational imaging and visualization, and has been created to underpin this new landscape.

THE MASSIVE FACILITY

MASSIVE has been established by Monash University, the Australian Synchrotron, the Commonwealth Scientific Industrial Research Organization (CSIRO), and the Victorian Partnership for Advanced Computing (VPAC) to support next-generation imaging and instrumentation. This facility provides computer hardware, software and expertise to drive research in the biomedical science, materials research, engineering, and neuroscience communities, and it stimulates advanced imaging research that will be exploited across a range of imaging modalities, including synchrotron x-ray and infrared imaging, functional and structural magnetic resonance imaging, x-ray computer tomography (CT), electron microscopy, and optical microscopy.

The MASSIVE project has a number of objectives. First, to provide a world-class imaging and visualization facility to research groups identified by the MASSIVE stakeholders. Second, to increase the uptake of imaging and visualization services by research groups using the Australian Synchrotron and by

Australian research groups more generally. Third, to increase the performance and capability of imaging and visualization systems, especially the on-line reconstruction of images generated by the Imaging and Medical Beamline (IMBL) at the Australian Synchrotron. And fourth, to increase the capabilities of research groups to use and develop imaging and visualization services.

MASSIVE is a unique Australian facility with a focus on fast data processing, including processing data "in-experiment," large-scale visualization, and analysis of large-cohort and longitudinal research studies. It provides this service within a national context of peak and specialized HPC facilities (**Figure 1**). The facility runs an instrument integration program to allow researchers to more easily process imaging data, and provides a high-performance managed interactive desktop environment providing access to common interactive analysis and visualization tools. MASSIVE offers Australian scientists access to two specialized computing facilities at Monash University and Australian Synchrotron with computer systems linked by a high-bandwidth communications link.

MASSIVE also manages a major nationally funded software infrastructure collaboration to make scientific tools, and in-particular neuroinformatics tools, available freely and cloud-ready. This collaboration, which is called the Characterization Virtual Laboratory, is composed of members of the Australian Characterization Council, the Australian Synchrotron, the Australian Nuclear Science and Technology Organization (ANSTO), the Australian Microscopy and Microanalysis Research Facility (AMMRF) and the National Imaging Facility (NIF), as well as Monash University, the University of Queensland, the Australian National University, and the University of Sydney. MASSIVE is participating in this project to support new imaging research disciplines in applying HPC, and to further develop the interactive analysis and visualization component of MASSIVE.

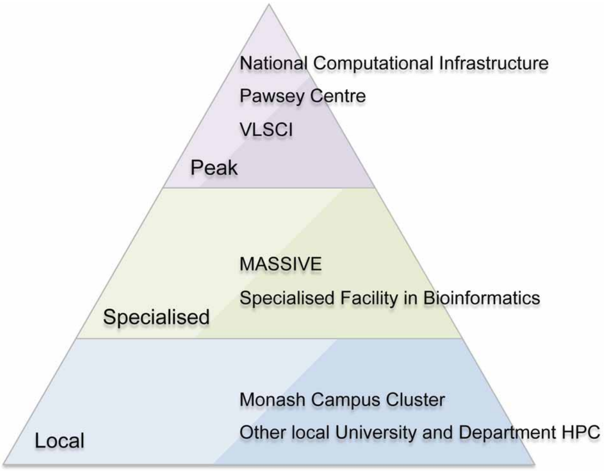

National Computational Infrastructure
Pawsey Centre
VLSCI
Peak
MASSIVE
Specialised Facility in Bioinformatics
Specialised
Monash Campus Cluster
Other local University and Department HPC
Local

FIGURE 1 | The Australian high performance computing (HPC) environment including peak (national) facilities, specialized national facilities, and local HPC facilities.

The total cost of MASSIVE exceeded AUD$5 million with additional contributions from the Australian Synchrotron, Monash University, CSIRO and VPAC, and is initially operational for three years until mid 2014. The MASSIVE facility is also part funded the National Computational Infrastructure (NCI) to provide imaging and visualization high performance computing facilities to the Australian scientific community. This agreement designates MASSIVE as the NCI Specialized Facility for Imaging and Visualization and allows researchers across Australia to access it based on merit allocation.

A Collaboration Agreement underpins the governance arrangements and includes a Steering Committee with an independent chair and members who are representatives of the partner organizations. The committee is guided by two Science Advisory Committees, which are the Synchrotron Science Advisory Committee and the Imaging and Visualization Advisory Committee. The facility provides an extensive program of user support and training on all aspects of high performance computing, and has an active outreach program to ensure that the MASSIVE stakeholders, Australian and international researchers, government and the broader community are aware of its benefits and achievements.

MASSIVE AND APPLICATIONS TO NEUROSCIENCE AND NEUROINFORMATICS

Advanced imaging instruments, including CT and MRI scanners and electron and optical microscopes, are capable of producing data at an incredible rate. As an example, the Australian Synchrotron Imaging Beamline is able to produce data at over 500 Mbytes/s. This introduces obvious challenges for researchers to capture, process, analyze, and visualize data in a timely and effective manner. Researchers are also increasingly eager to perform data analysis "in-experiment" so that they can make appropriate decisions in real-time. MASSIVE provides real-time imaging support as follows:

- Integration of the data sources (the instruments) with the data storage and data processing engines (MASSIVE or other HPC facility) including an instrument integration support program for this purpose; and
- Provision of a common desktop environment for data processing, analysis, and visualization that is integrated with the HPC capability, and allows researchers to access their data through an environment that supports both the desktop and HPC tools they use to process their data.

This configuration results in researchers moving their data only once, automatically during data capture, with subsequent processing, analysis, and visualization performed centrally on MASSIVE. The outcome is that MASSIVE is able to support communities that have not traditionally used HPC computing.

MASSIVE currently supports over 25 Australian neuroinformatics research projects that include researchers who are:

- Undertaking large-cohort studies and longitudinal studies such as the ASprin in Reducing Events in the Elderly (ASPREE) study (Nelson et al., 2008) and the IMAGE-HD Huntington's disease study (Georgiou-Karistianis et al., 2013);

- Processing, analysing, and viewing data generated by advanced imaging equipment, including the Australian Synchrotron Imaging Beamline, new generation Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and other techniques;
- Applying computer tomography techniques or volume visualization and analysis techniques;
- Applying advanced image processing, image analysis, or visualization techniques, or undertaking research in these fields; and
- Developing modeling and simulation applications, in particular applications that are suited to fast file system access or GPU hardware.

COMPUTING INFRASTRUCTURE FOR NEUROINFORMATICS

Scientific applications of HPC, cloud and grid computing have been thoroughly documented and computing is considered an essential scientific tool (Foster and Kesselman, 2003). A number of specialized undertakings for bioinformatics, and more specifically neuroinformatics, have been very successful and deserve particular comment.

The Biomedical Informatics Research Network (BIRN) (Grethe et al., 2005) is an infrastructure to help communities build virtual organizations, and includes support for data sharing, security, authentication and authorization, and scientific workflows. The Functional Bioinformatics Research Network (fBIRN) is a specific application of BIRN for neuroimaging, allowing researchers to calibrate and collect fMRI data across sites, and manage and analyse that data (Greve et al., 2010). Similarly, CBRAIN and GBRAIN (Frisoni et al., 2011) are an online collaborative web platform for neuroimaging allowing users to access a wide range of participating HPC resources, in Canada and across the globe.

A number of projects provide dedicated HPC access and support to neuroimaging researchers. These include the NeuGrid Redolfi (Redolfi et al., 2009) and it's successor N4U (Haitas and Glatard, 2012), and the NeuroScience Gateway (NSG) (Sivagnanam et al., 2013). All three projects provide web-based mechanisms for data management and processing and analysis on HPC systems, and specialized support for neuroimaging.

In addition there are a number of online and desktop workflow environments that are being applied to general science and specific bioinformatics and neuroinformatics purposes. These include Galaxy (Giardine et al., 2005), the LONI Pipeline (Rex et al., 2003), Kepler (Ludäscher et al., 2006), and Soma-workflow (Laguitton et al., 2011). These projects all provide mechanisms to interface with high performance computing resources. Nipype (Gorgolewski et al., 2011) is a workflow for interfacing with a range of neuroinformatics packages, allowing users to easily compare algorithms across packages. PSOM (Bellec et al., 2012) is a workflow engine for Octave and Matlab developed for neuroimaging.

The Blue Brain Project (Markram, 2006) is undertaking to simulate the brain on a HPC. The project commenced by undertaking to simulate a cellular-level model of a 2-week-old rat somatosensory neocortex based on captured microscopy data, specifically targeting the IBM Blue Gene HPC platform. This project, has now evolved into the broader Human Brain Project (HBP, 2012), which is discussed in Section Large-scale International Initiatives.

MASSIVE shares many of the fundamental goals of these projects—to provide neuroscience researchers with access to high performance computing capabilities and data management. However, our project differs in a number of ways:

- Integration of scientific instrumentation is a key feature of the project, allowing scientists to perform sophisticated processing immediately after data capture, and in some cases performing data processing as part of the experiment (Section Instrument Integration Program);
- Easy access for non HPC-experts is important to support the broad neuroscience community. Many of the projects discussed approach this problem by providing access to web portals or workflow environments. MASSIVE has decided to take the approach of providing a remote desktop (Section Massive Interactive Software Environment), which has proved effective in helping researcher transition from their personal desktop to a HPC environment. It also alleviates the need to wrap tools in a web front-end and means that a vast range of desktop tools can be supported on the systems.
- We are actively developing the MASSIVE software stack to the cloud (Section Neuroinformatics in the Cloud) which will make MASSIVE more accessible to a wider range of neuroscientists.

INFRASTRUCTURE

HARDWARE

MASSIVE consists of two interconnected computers, M1, and M2 respectively, that operate at over 5 and 30 teraflops[1] respectively, using traditional CPU processing, and accelerated to over 50 and 120 teraflops[1], respectively, using co-processors. M1 and the first stage of M2 were made available to Australian researchers in May 2011. The computers are connected using a dedicated connection for fast file transfer and common management. A summary of the technical specifications of the two systems and the hardware configuration of the two computers, including the GPU coprocessors and the parallel file systems, are given in **Table 1**.

GPUs have proved an important part of the MASSIVE environment. Key applications, including the X-TRACT (Gureyev et al., 2011) CT reconstruction software, have been parallelized to take advantage of the GPUs. This has been critical to performing fast processing of data in a near real-time fashion as discussed in Section Instrument Integration Program. Moreover, GPUs have become an important developmental technology for the research community and MASSIVE has supported a number of projects to successfully port imaging analysis code to the GPU environment. Section GPU reconstruction of quantitative magnetic susceptibility maps of the human brain describes a specific example of

[1] Theoretical performance of the systems.

Table 1 | Technical specifications of the MASSIVE high performance computing system.

M1 AT THE AUSTRALIAN SYNCHROTRON
42 nodes (504 CPU-cores total) in one configuration:
42 nodes with 12 cores per node running at 2.66 GHz
48 GB RAM per node (2016 GB RAM total)
2 NVIDIA M2070 GPUs with 6GB GDDR5 per node (84 GPUs total)
153 TB of fast access parallel file system
4x QDR Infiniband Interconnect
M2 AT MONASH UNIVERSITY
118 nodes (1720 CPU-cores total) in four configurations:
32 nodes with 12 cores per node running at 2.66 GHz
48 GB RAM per node (1536 GB RAM total)
2 × NVIDIA M2070 GPUs with 6 GB GDDR5 per node (64 GPUs total)
10 nodes with 12 cores per node (visualization/high memory configuration)
192 GB RAM per node (1920 GB RAM total)
2 × NVIDIA M2070Q GPUs with 6 GB GDDR5 per node (20 GPUs total)
56 nodes with 16 cores per node running at 2.66 GHz
64 GB RAM per node (3584 GB RAM total)
2 × NVIDIA K20 (9 nodes—18 GPUs total)
2 × Intel PHI (10 nodes—20 coprocessors total)
20 nodes with 16 cores per node running at 2.66 GHz
128 GB RAM per node (2560 GB RAM total)
2 × NVIDIA K20 (40 GPUs total)
345 TB of fast access parallel file system
4 × QDR Infiniband Interconnect

Combined the M1 and M2 have 2,224 CPU-cores.

the application of GPUs to Quantitative Susceptibility Mapping (QSM). Importantly, the GPU capability allows MASSIVE to provide good support for interactive visualization, including through the MASSIVE Desktop (Section MASSIVE Interactive Software Environment) and through parallel rendering tools such as Paraview (Henderson et al., 2004).

Both M1 and M2 have a GPFS (Schmuck and Haskin, 2002) file system that is capable of a combined 5 GB+ per second write speed. This capability has proved essential to support both the fast capture of data from instruments, and file system intensive image processing workloads. Section Instrument Integration Program discusses the importance of the file system to support large-scale and real-time CT reconstruction image processing applications.

INSTRUMENT INTEGRATION PROGRAM

MASSIVE has a dedicated program for the integration of imaging instruments with high performance computing capability (**Figure 2**, **Table 2**) that gives scientists the ability to use complex and computationally demanding data processing workflows within minutes of acquiring image datasets. Instruments integrated with MASSIVE that are of particular interest for neuroscience research include MRI and CT equipment at Australian National Imaging Facility locations across Australia, and for near real-time CT image reconstruction on the Imaging Beamline at the Australian Synchrotron.

The instrument integration program allows scientists to visualize and analyse collected data as an experiment progresses or shortly after it completes, thereby integrating processing, analysis and visualization into the experiment itself. In particular, groups that are imaging live anesthetized animals must be able to establish whether a previous scan has successfully produced the desired data before proceeding with the next step of the experiment. These experiments are typically time-critical as there is limited instrument availability once an experiment has commenced. In many cases the images captured by detectors at the Imaging Beamline are very large and necessitate the rapid movement of TB data sets for processing. These constraints dictate that significant computing power is required on demand and that the computer is tightly coupled to the instruments and readily available to the researchers.

Data management at Monash Biomedical Imaging

Neuroimaging studies, especially multi-modal, longitudinal studies of large cohorts of subjects, generate large collections of data that need to be stored, archived, and accessed. MRI based studies can easily accumulate terabytes of data annually and require integration of HPC and informatics platforms with the imaging instrumentation. Integrated systems that combine data, meta-data, and workflows are crucial for achieving the opportunities presented by advances in imaging facilities. Monash University hosts a multi-modality research imaging data management system that manages imaging data obtained from five biomedical imaging scanners operated at Monash Biomedical Imaging (MBI) (**Figure 3**). In addition to Digital Imaging and Communications in Medicine (DICOM) images, raw data and non-DICOM biomedical data can be archived and distributed by the system. Research users can securely browse and download stored images and data, and upload processed data via subject-oriented informatics frameworks (Egan et al., 2012) including the Distributed and Reflective Informatics System (DaRIS) (Lohrey et al., 2009; DaRIS, 2013), and the Extensible Neuroimaging Archive Toolkit (XNAT) (Marcus et al., 2007).

DaRIS is designed to provide a tightly integrated path from instrument to repository to compute platform. With this framework, the DaRIS system at MBI manages the archiving, processing, and secure distribution of imaging data (with the ability to handle large datasets) acquired from biomedical imaging scanners and other data sources. This ensures long-term stability, usability, integrity, integration, and inter-operability of imaging data. Imaging data are annotated with meta-data according to a subject-centric data model and scientific users can find, download, and process data easily. DaRIS users can export their data directly into their MASSIVE project environment for analysis.

Recent enhancement of DaRIS (Killeen et al., 2012) provides for the management and operation of workflows (using the Nimrod and Kepler technologies) with input and output data managed by DaRIS. In this way, large subject-cohort projects can robustly process (and re-process) data with attendant enhanced data provenance. Current DaRIS enhancements are focusing on additional efficient data inter-operability capabilities so that

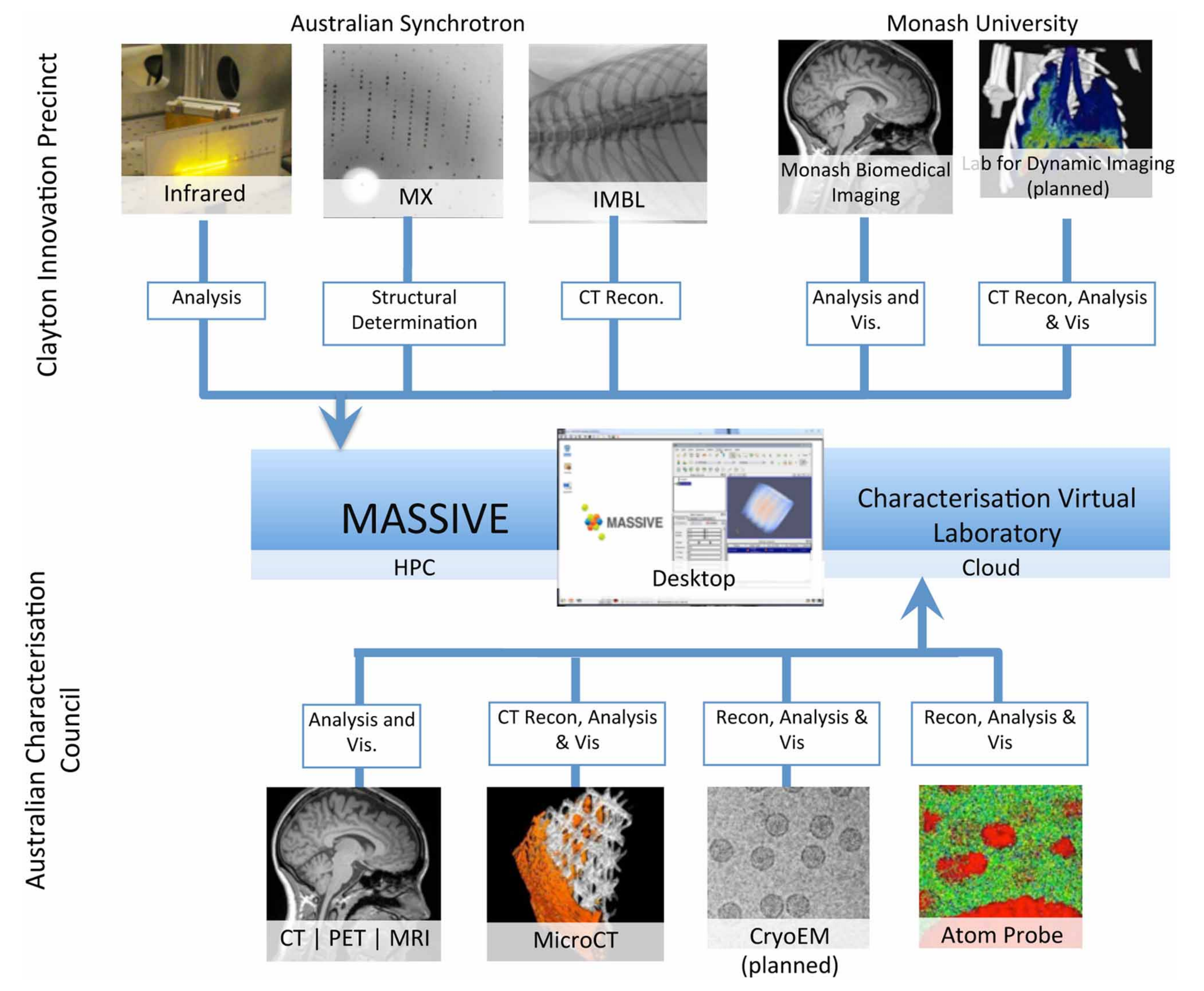


FIGURE 2 | A schematic of the integration of access to imaging instrumentation from the MASSIVE desktop and the Cloud via the Characterization virtual laboratory.

researchers can access their managed data when and where they need it.

Australian synchrotron imaging beamline CT reconstruction

The MASSIVE computers have been integrated with a number of beamlines at the Australian Synchrotron, and provide a range of data processing services to visiting researchers. These include: near real-time image reconstruction at the IMBL, near-real time automated structural determination at the Macromolecular Crystallography beamline, microspectroscopy at the Infrared beamline, data analysis at the Small and Wide Angle Scattering beamlines, and image analysis at the X-ray Fluorescence Microprobe Beamline. These techniques are being applied to a range of biomedical sciences and neuroimaging applications.

The IMBL has a capability of near real-time high-resolution CT imaging of a range of samples, including high-resolution phase-contrast x-ray imaging of biomedical samples, animal models used in neuroscience experiments, and engineering materials. The beamline is 150 meters long, with a satellite building that includes a medical suite for clinical research as well as extensive support facilities for biomedical and clinical research programs. Two detectors based on pco-edge cameras are available for use. Typical data acquisition times are dependent upon the chosen x-ray energy and detector resolution and vary approximately between 10 and 60 min for a complete CT scan. When imaging data is acquired at the maximum detector resolution and at 50 frames per second, the data rate is $2560 \times 2160 \times 2$ byte $\times$ 50 fps = 527.34 Mbytes/s. **Figure 4** illustrates the architecture of the IMBL CT Reconstruction service.

In order to control the data collection and optimize the experimental conditions at IMBL, scientists must be able to visualize collected data in near real-time as the experiment is in progress. In particular, groups that are imaging live anesthetized animals

Table 2 | The computational systems and file system access associated with the imaging instrumentation integrated with MASSIVE and the Characterization Virtual Laboratory.

Instrument	Capture method	Service	Scientific capability
INTEGRATED			
Imaging and Medical Beamline	File system integration	GPU processing, parallel FS, and interactive visualization	CT reconstruction and visualization
Macromolecular Crystallography Beamline	File system mount	Compute	Structural determination
Infrared Beamline		Compute	Signal correction
X-ray Fluorescence Microprobe Beamline		Parallel FS and interactive visualization	Analysis
Small Angle and Wide Angle X-ray Scattering		Compute	Modeling
CT and MRI Imaging Instruments	DaRIS	GPU processing, parallel FS, and interactive visualization	Data capture, analysis, and visualization
Electron Microscopes	Tardis	Parallel FS, cloud computing, and interactive visualization	Data capture, analysis, and visualization
PLANNED OR IN PROGRESS			
Biomedical X-ray sources	File system mount	GPU processing, parallel FS, and interactive visualization	CT reconstruction and visualization
Atom Probes	Tardis	Cloud computing and interactive visualization	Analysis
Electron Microscopes	Tardis	GPU processing, parallel FS, and interactive visualization	Structural determination and visualization
Micro-CT X-ray sources	File system mount		CT reconstruction and visualization
Soft X-ray Beamline			CT reconstruction

often need to establish whether a previous scan has successfully produced the desired data before proceeding with the next step of an experiment. The experiments are typically time-critical as the window of the experiment once begun is short. The image datasets captured by detectors at the IMBL require the manipulation of data sets in the terabyte range. These experimental constraints dictate that significant computing power is tightly coupled to the experimental detectors and available on-demand.

CT data sets collected at IMBL are typically tens of GB per sample consisting of typically 1200–2400 projection images that can be acquired from a single sample in less than 1 min. The X-TRACT package on M1 is the primary software available to users for reconstruction of CT data, including phase-contrast CT as implemented at IMBL (Goscinski and Gureyev, 2011; Gureyev et al., 2011). Usage of MASSIVE for CT reconstruction via X-TRACT is offered during the synchrotron experiments and also via remote access up to 6 months after an experiment has been completed, allowing researchers to process and analyse captured data remotely. The CT reconstruction service has been in production since November 2012.

The X-TRACT customized CT image reconstruction software is parallelized for the MASSIVE GPU and parallel file system architecture. X-TRACT is an application for advanced processing of X-ray images that also provides multiple image processing tools. In particular, there is extensive functionality for X-ray CT image processing, including multiple methods for CT reconstruction and X-ray phase retrieval and simulation of phase-contrast imaging. Additionally, a large number of operations such as FFT, filtering, algebraic, geometric, and pixel value operations are provided. X-TRACT has been designed to fully leverage the processing capabilities of modern hardware such that computationally intensive operations utilize multiple processors/cores and GPU's where available to increase performance. The X-TRACT software has been adapted for use on HPC cluster infrastructure, and has been optimized for the MASSIVE systems, to enable it to process multi-TB datasets produced at synchrotron light sources that are unable to be processed on standalone desktop machines.

To demonstrate the importance of the file system capability in particular, benchmarking of X-TRACT on the M1 cluster has been performed using the Gridrec CT reconstruction algorithm (Rivers and Wang, 2006) for multi-TB datasets. The total reconstruction time and IO time as a proportion of the runtime for a set of CT reconstructions is shown in **Figure 5** as a function of the number of CPU cores. The input dataset consisted of 8192 pre-processed sinogram files (total ∼1.5 TB), and the output was an 8192^3 pixel dataset (total ∼2 TB). The results demonstrate that IO represents a significant proportion of the overall running time—particularly beyond 36 CPU-cores. We are currently investigating the refinement of the HPC based CT processing workflow to reduce the high proportion of IO time which is currently the major performance bottleneck.

MASSIVE INTERACTIVE SOFTWARE ENVIRONMENT

MASSIVE provides users with highly accessible high-performance scientific desktop—an interactive environment for analysis and visualization of multi-modal and multi-scale data (**Figure 6**). This environment provides researchers with access to a range of existing tools and software, including

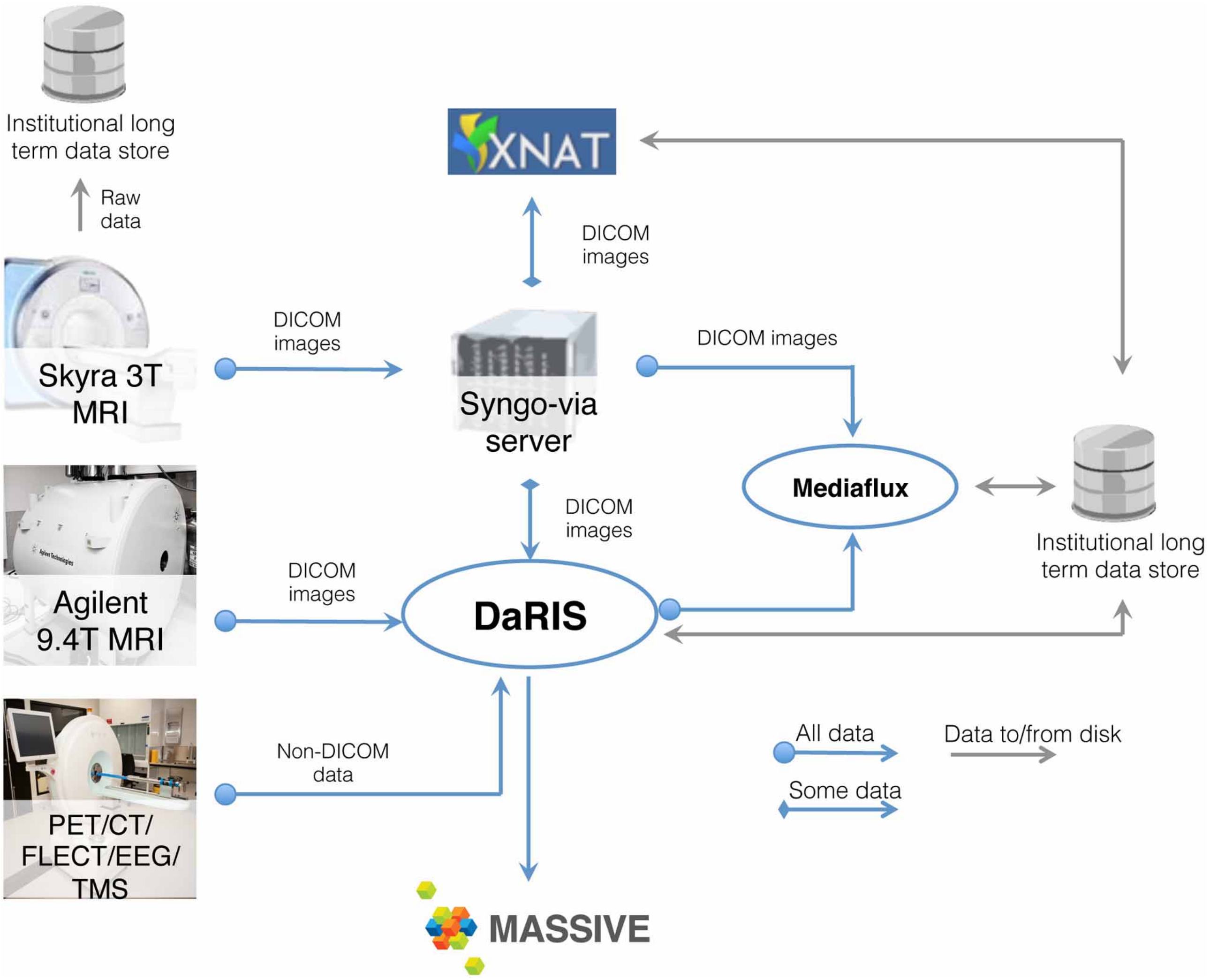


FIGURE 3 | Schematic of the neuroscience image data flow from Monash Biomedical Imaging and the computational processing performed on M2.

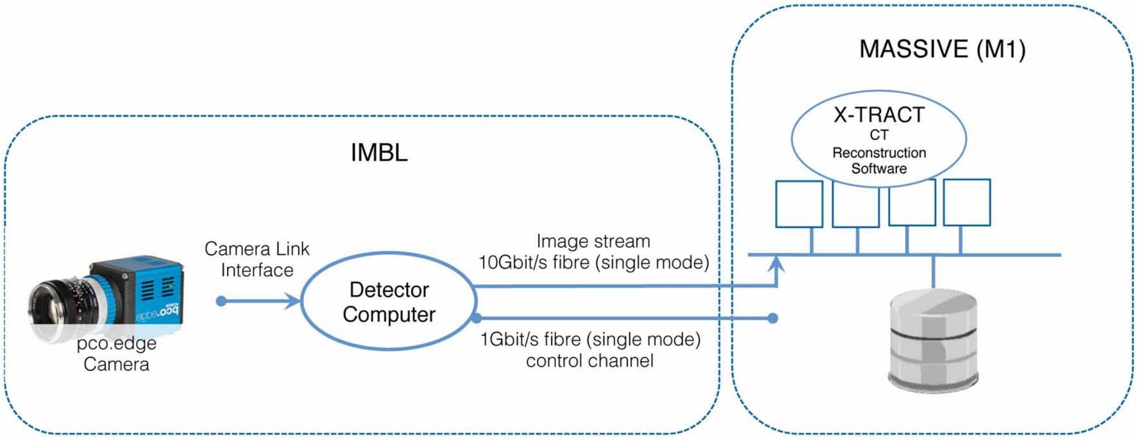

MASSIVE (M1)
X-TRACT
CT Reconstruction Software
IMBL
Camera Link Interface
Detector Computer
Image stream 10Gbit/s fibre (single mode)
1Gbit/s fibre (single mode) control channel
pco.edge Camera

FIGURE 4 | Schematic of the architecture of the IMBL CT Reconstruction service provided on M1.

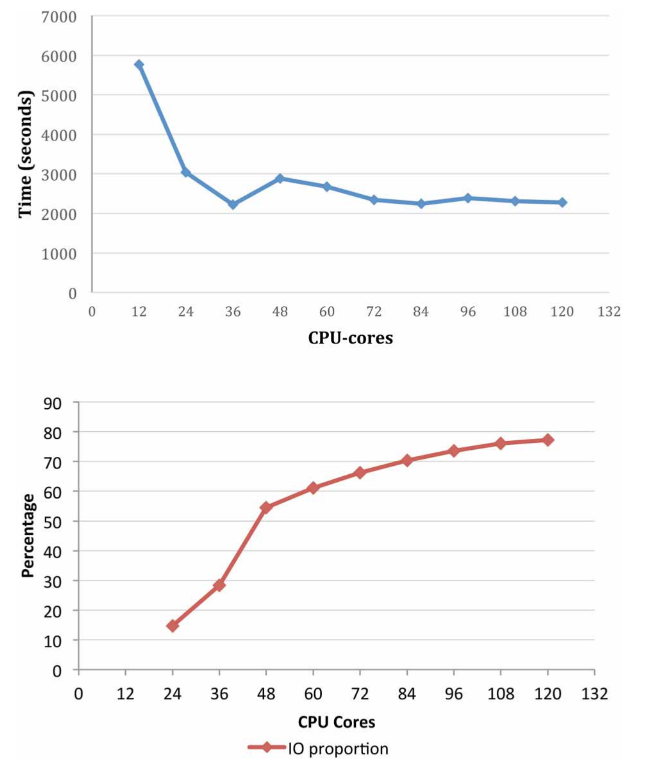


FIGURE 5 | The total reconstruction time for CT reconstruction of an 8912^3 dataset (top) and IO time as a proportion of runtime (bottom) on M1 as a function of the number of CPU cores.

commercial and open-source neuroinformatics applications. Common neuroimaging applications such as FSL (Smith et al., 2004) and SPM (Friston et al., 1994) have been integrated into the desktop to allow users to submit HPC jobs without specific HPC knowledge. The continual growth in data and study sizes increasingly necessitates the analysis and rendering of data at the location where the data is stored. Furthermore, performing analysis and visualization on a central facility greatly increases the efficiency and flexibility for researchers to access high performance hardware, including fast file systems and GPUs. Together with the MASSIVE Instrument Integration program, the desktop provides a fully integrated environment that allows researchers to view and analyze images shortly after the imaging data has been acquired.

The scientific desktop allows MASSIVE users to access a wide range of analysis tools without rewrapping or reengineering of the tools. The remote desktop has been built using CentOS running the KDE or Gnome desktop environment. For remote access, the desktop uses an open source VNC implementation, TurboVNC (http://www.virtualgl.org/), as it supports remote hardware accelerated rendering and clients on all three major platforms: Windows, Mac, and Linux. Network latency and bandwidth using the Australian academic research network (AARNET) is sufficient to support TurboVNC across the Australian imaging research community and the MASSIVE desktop is commonly accessed from every major city in Australia. The MASSIVE desktop supports a simple launcher called Strudel (short for Scientific Desktop Launcher) that automates the steps to access a desktop session. The Launcher launches an interactive visualization job on the MASSIVE system, and connects using TurboVNC using a secure SSH connection. The launcher is provided for all three major desktop platforms. It is configurable to other facilities and is being applied at other HPC facilities in Australia. It is available open source (Section Software and System Documentation).

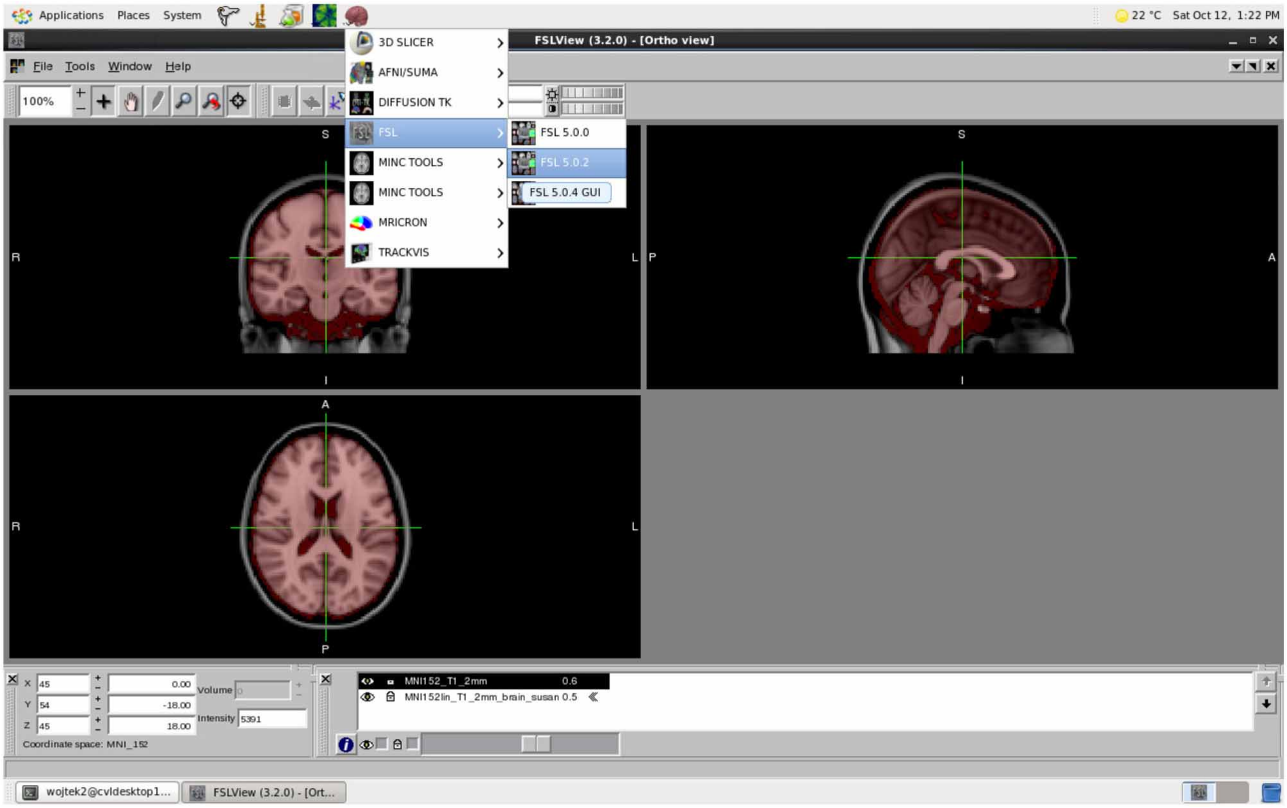
NEUROINFORMATICS IN THE CLOUD

To make imaging tools more accessible to the scientific community, MASSIVE is a key participant in the Australian Characterization Virtual Laboratory (CVL) project that is funded under the National eResearch Collaboration Tools and Resources (NeCTAR) project (www.nectar.org.au). The NeCTAR CVL project is an open source project aimed at porting key scientific imaging applications to the cloud with a particular focus on neuroinformatics tools (Goscinski, 2013).

The CVL has developed a managed desktop environment, based on the MASSIVE Desktop, including the Neuroimaging Workbench to support the neuroscience imaging community. The CVL environment provides access to the MASSIVE file system and job queues and is supporting further expansion of the instrument integration program (**Figure 2**). The Neuroimaging Workbench has integrated workflow and database systems to allow researchers using instruments managed by the Australian National Imaging Facility (NIF) to process and manage large neuroimaging datasets. The Australian NIF is a national network of universities and biomedical research institutes that provides key biomedical imaging instruments and capabilities for the Australian research community.

Neuroinformatics tools in the cloud have great potential to accelerate research outcomes. The Neuroimaging Workbench includes a project for registration of multi-modal data brain data for the Australian Mouse Brain Mapping Consortium (Richards et al., 2011). Ultra-high resolution 15 um MRI and micro-CT images from excised tissue, can be registered with 3D reconstructions of histological stained microscopy sections. The registered datasets enable the MRI and CT images to be correlated at both the microscopic (cellular) and macroscopic (whole organ) scales. A mouse brain atlas that combines ultra-high resolution MRI and histological images has wide ranging application in neuroscience. However, image registration of 3D microscopy and MRI datasets requires immense computational power as well as a range of specialized software tools and workflows, the developed workflow is applicable to all small animal atlas building efforts.

A major objective of the CVL Neuroimaging Workbench is to increase the efficiency for the neuroimaging community to undertake complex image processing and analyses for large and longitudinal scale studies. The integration of key imaging instruments across multiple nodes of NIF is allowing neuroimaging researchers to efficiently stage data to the cloud for processing on HPC facilities. The workbench provides researchers with simple and free access to a high performance desktop environment, that contains a fully configured set of neuroimaging tools for analysis and visualization, that may obviate the need for high-end desktop workstations that are currently replicated across many neuroimaging laboratories.



X 45 0.00
Y 54 -18.00
Z 45 18.00
Volume 0
Intensity 5391
Coordinate space: MNI_152

MNI152_T1_2mm 0.6
MNI152lin_T1_2mm_brain_susan 0.5

FIGURE 6 | The MASSIVE Desktop environment showing FSLView and a range of neuroinformatics tools available through the menu.

SOFTWARE AND SYSTEM DOCUMENTATION

User system documentation for MASSIVE and the infrastructure developed under the Characterization Virtual Laboratory is available publically (www.massive.org.au). In addition, system documentation is available on request. Software developed under the Characterization Virtual Laboratory to support remote desktops and the neuroimaging workbench is available open source as they enter beta release (www.massive.org.au/cvl).

APPLICATIONS IN NEUROSCIENCE IMAGING

APPLICATION TO HUMAN BRAIN IMAGING IN HUNTINGTON's DISEASE

The IMAGE-HD study is an intensive multi-modal MRI longitudinal study in Huntington's disease (Georgiou-Karistianis et al., 2013). The IMAGE-HD study is investigating the relationships between brain structure, microstructure and brain function with clinical, cognitive and motor deficits in both pre-manifest and symptomatic individuals with Huntington's disease. Structural, functional, diffusion tensor, and susceptibility weighted MRI images have been acquired at three time points in over 100 volunteers at study entry, and after 18 and 30 months. This data is managed in the DaRIS environment. Multi-modal imaging was used to identify sensitive biomarkers of disease progression for recommendation in future clinical trials. The multi-modal imaging results have demonstrated evidence of differential rates of change in both Huntington's disease groups across a range of imaging measures with changes detected up to 15 years before the onset of symptoms (Domínguez et al., 2013; Gray et al., 2013).

The MASSIVE desktop has been used to undertake the computational imaging analyses of the structural, diffusion, and functional MRI data acquired in the IMAGE-HD study. Longitudinal diffusion tensor imaging datasets have been analyzed using deterministic (trackvis.org) and probabilistic (Behrens et al., 2007) tractography tools that have been recoded for the MASSIVE GPU and made available via the desktop. Network level brain dysfunction in axonal fiber-connectivity in HD has been analyzed using MASSIVE (Poudel et al., 2013), as well as resting-state fMRI data analyses using graph theoretical methods (Zalesky et al., 2010). The desktop is used to run semi-automated analysis pipelines for tracking longitudinal changes in structural connectivity, diffusivity in white matter, and functional connectivity in HD. The desktop is also being to used to develop combined analyses of fMRI and DTI datasets in order to understand the relationships between brain functional and microstructural deficits in Huntington's disease.

GPU RECONSTRUCTION OF QUANTITATIVE MAGNETIC SUSCEPTIBILITY MAPS OF THE HUMAN BRAIN

Quantitative Susceptibility Mapping (QSM) (Duyn et al., 2007; Liu et al., 2009) is a technique used in MRI to measure the magnetic susceptibility of tissue, which in turn relates to the paramagnetic content of the tissue. Diffusion guided QSM (dQSM) (Ng, 2013) is a new technique that uses diffusion MRI data to improve the modeling of magnetic susceptibility at each position in the image, but it is a computationally challenging problem, requiring the inversion of a multi-terabyte matrix. Diffusion guided QSM treats the magnetic susceptibility effect of each image voxel as isotropic (Liu et al., 2011) or axial (Lee et al., 2012) depending on the fractional anisotropy (FA) in corresponding diffusion-weighted images. The computation of the matrix formulation of the problem using the Landweber iteration (LI) method is prohibitively expensive on central processing unit (CPU) cores. Acceleration of the algorithm by utilizing graphics processing unit (GPU) cores is necessary to achieve image computation times practical for research use today, and for clinical application in the near future. The dQSM problem is suited to the GPU for the reason that the elements of the matrix in the Landweber iteration formulation can be computed on-demand; without this ability the problem would be intractable on GPUs. By computing the elements of the matrix on-the-fly using the MASSIVE GPU architecture the time for computation of QSM images has been reduced by a factor of 15.

Several attributes of the Landweber iteration method applied to the dQSM problem make it particularly suitable to the GPU architecture. Computing the solution requires iteratively multiplying very large matrices, which are computed on-the-fly from smaller input buffers, with vectors of voxel input data and adding the result to the previous values. Each iteration is an Order (N^2) problem with a high computational load of calculating the matrix elements that extensively uses multiply-then-add that allows fused multiply-add instructions. The conveniently contiguous access to most of the read/write data vectors by parallel computational threads enables better cache performance and reduced global memory read/write overheads. By computing the elements of the matrix on-the-fly and optimizing to best use the MASSIVE GPU architecture, the time for computation of QSM images has been reduced by a factor of 15.

The reference CPU solution uses an MPI parallel processing paradigm that already provides a domain decomposition. This decomposition was applied to the GPU implementation to split separate sections of the problem over a number of GPUs in an additional layer of parallelism. The MASSIVE architecture provides two NVIDIA Tesla M2070 or K20 GPUs per compute node along with 12 CPU cores. The fast interconnect between nodes enabled excellent scaling on the multiple GPU code with minimal communication overhead even when computed on up to 32 GPUs over 16 nodes. Current work involves a more intelligent load balancing of the work across multiple GPUs and potentially separating the problem into white-matter voxels (which require the LI technique and therefore the huge level of compute power the GPU provides), and other voxels which can be computed using a fast Fourier transform based technique. This would permit utilization of the CPU cores that sit idle while the GPU computation is performed.

The dQSM method implemented on the MASSIVE GPU architecture demonstrates greater accuracy in susceptibility estimation results compared to methods based solely on a spherical diffusion mode. The major disadvantage is the very long computation time, which makes the method challenging for routine research and clinical applications. Algorithmic improvements and the growth in compute capability of GPUs together with the further speed-up of the GPU implementation being undertaken, is expected to enable clinically-relevant post-processing times (less than 30 min). Using multi-component models of tissue structures to estimate susceptibility effects will provide more accurate results with further improvements in implementation of the dQSM algorithm.

DIGITAL ATLASING OF THE MOUSE BRAIN

The mouse is a vital model to elucidate the pathogenesis of human neurological diseases at a cellular and molecular level. The importance of the murine model in neuroscience research is demonstrated by the multitude and diversity of projects including the Allen Brain Atlas (brain-map.org), Waxholm Space (waxholm.incf.org) developed under the auspices of the International Neuroinformatics Coordinating Facility, the Mouse Brain Library (MBL) (mbl.org) and the Mouse Brain Architecture Project (MBAP) (brainarchitecture.org). Many research groups use non-invasive MRI to structurally map the murine brain in control and disease model cohorts. Until recently, the construction of mouse brain atlases has been relatively restricted due to the variety of sample preparation protocols and image sequences used, and the limited number of segmented brain regions.

The Australian Mouse Brain Mapping Consortium (AMBMC) has recently developed an ultrahigh resolution and highly detailed MRI-based mouse brain atlas (Richards et al., 2011; Ullmann et al., 2012). The AMBMC atlas has initially concentrated on five primary brain regions, the hippocampus, cortex, cerebellum, thalamus, and basal ganglia and has recently published a segmentation guide and probabilistic atlas for over 200 structures. MRI data from 18 C57BL/6J mice was acquired at 30 μm^3 resolution, averaged to create a single image at a resolution of 15 μm^3, and placed in the stereotaxic Waxholm space. The components of the brain were delineated, on the bases of differences in signal intensity and/or their location in reference to landmark structures. A digital atlas containing over 200 structures with mean region volumes, T2*-weighted signal intensities and probability maps for each of structure was generated for use as a detailed template for cross modality applications (see www.imaging.org.au/AMBMC).

These components have been integrated and made available through the Neuroimaging Workbench (Janke, 2013).

DISCUSSION AND FUTURE

There are a number of major trends that will influence MASSIVE, both under its current project plan and in the future. This includes technological trends, capabilities such as visualization, and major international initiatives.

MASSIVE CHALLENGES

Our experience developing and managing the MASSIVE systems has highlighted a number of noteworthy challenges.

The MASSIVE systems cannot be managed in the same way as a more traditional HPC facility where computer utilization is a key measure of success. Because we commonly provide access to compute in a near-realtime or interactive manner, we must keep a proportion of the systems available and waiting for instrument processing or desktop sessions. We aim for CPU-core utilization of around 70%, as opposed to more traditional systems that are able to achieve between 90 and 100% utilization. We are experimenting with strategies such as dynamic provisioning of nodes and short running jobs to fill idle time.

Interactive desktop sessions on our facility run on a dedicated node. Thus, users have access to two CPU processors running between 8 and 12 cores, and up to 192 GB of memory. We do not allow multiple users onto a single desktop node, because a user can inadvertently affect other users. For example, by launching a multi-core application. However, a significant proportion of desktop users do not require access to the full technical capabilities. For example, a user that is using an image viewer to examine a large dataset might only require one CPU-core. The result is wasted computing resources. Our long-term plan to solve this problem is to host desktop sessions in virtual machines that will be provisioned at specific sizes and capabilities. Using virtual machines allows us to completely isolate multiple users of a single desktop and ensure a good user experience. In our early experience with provisioning on the cloud (Section Neuroinformatics in the Cloud) the overhead imposed by a virtual machine is acceptable, but fast access to file systems needs to be carefully considered.

Our most significant challenge is not technical but relates to user support. In a traditional HPC environment users will be accustomed to submitting jobs to a queue and checking back for their results. In an interactive environment, small changes to performance and accessibility have a strong effect on user experience. Moreover, users require fast response to problems—particularly considering issues with the computing system can have a major effect a physical experiment. Our solution to this problem has been to ensure that have adequate expert staff who are able to quickly triage and prioritize problems.

TRENDS IN SCIENTIFIC COMPUTING

A major trend in HPC has been the application of GPU technology, developed primarily to support the gaming market, to enable fast parallel processing. This has continued to be driven by the development of new architectures, such as the Intel Phi.

Likewise, the trend toward centralized cloud hosting, and the competition between major cloud vendors has created a landscape where hosting applications in the cloud is a very economical solution, whilst still providing a high degree of control to customize a solution to a particular science question. Early cloud hardware offerings lacked specialized hardware, such as GPUs or high performance interconnects. However, cloud computing providers are increasingly providing these capabilities, including Amazon (Ekanayake and Fox, 2010) (Amazon, 2013). In addition, the development of open source cloud computing middleware, such as OpenStack (OpenStack, 2013), allows a broader range of providers to offer cloud solutions and increases the availability of specialized services—such as parallel hardware or scientific applications. In particular, through the NeCTAR project, a number of major Australian Universities are developing an OpenStack federated multi-node cloud for the research community (NeCTAR, 2013). The CVL project is hosted on this environment allowing it access to GPUs and, in the future, a low latency and high bandwidth connection to MASSIVE. The Neuroimaging Tools and Resources Clearinghouse (NITRC) (Buccigrossi et al., 2007) Computational Environment (NITRC, 2013), is an analogous project that, like the CVL, provides a cloud platform preconfigured for neuroinformatics. This allows any neuroscientist to easily access the latest tools running on the Amazon cloud for between $0.02 and $3.10 per hour depending on the hardware configuration.

These trends in computing are creating a landscape where cloud hosting of scientific applications—including interactive desktop applications—will become a feasible, economical, and powerful solution. MASSIVE is supporting this trend by porting neuroimaging applications to the cloud through the CVL project, and integrating key Australian instruments, including the IMBL and imaging equipment through the NIF.

VISUALIZATION FOR NEUROINFORMATICS

Understanding and visualizing information is a hurdle for researchers who generally work with 2D screens and rarely use 3D displays. Advances in research imaging technology has dramatically increased the volume and complexity of research data that is routinely collected. New virtual reality technologies now provide the possibility of panoramic 320° visual displays that match human visual acuity, and provide visualization opportunities for exploring, and understanding the complexity of neuroscience data, in particular human brain imaging data. The next generation of neuroscience discoveries underpinned by virtual reality technologies and advanced computational approaches have the potential to initiate a new discipline of visualization led scientific discovery in neuroscience. MASSIVE is collaborating with a unique Australian immersive visualization facility, the Monash University CAVE2 facility (CAVE2, 2013), to allow researchers to visualize MASSIVE 2D and 3D data in an immersive environment. The direct integration of the MASSIVE Desktop with the CAVE2 display facility, including support for 3D display from applications the MASSIVE users are already familiar with, is a key objective for the initial operating period of the CAVE2.

Scientists are increasingly applying a systems approach to understanding the human brain—coupling multiscale models to develop an understanding of how models work together, how effects propagate through systems and how high-level outcomes are constructed from fundamental physics and chemistry. There is a desire to provide mechanisms for interacting with and steering of simulations to understand emergent properties. In particular, the Human Brain Project (HBP, 2012; Markram, 2012) will develop mechanisms to gain visual feedback, steer simulations, and interrogate simulated models as if they were a real biological sample. New visualization tools for easily interacting with computational models, large-scale simulations, and big data are important to ensure HPC is easily accessible to the neuroscience community.

LARGE-SCALE INTERNATIONAL INITIATIVES

Several large-scale international brain research initiatives are now underway in both the US and Europe to accelerate our understanding of the brain and its diseases and disorders. The

Human Brain Project (HBP) has been funded with the aim to take advantage of the convergence between ICT and biology to model the brain in a single multi-level system. The HBP will use supercomputers to build and simulate brain models with unprecedented levels of biological detail, and use data from new sequencing and imaging technologies, cloud technology, and neuroinformatics. The neuroinformatics community is already working closely with the large-scale initiatives to ensure collaboration on computational neuroscience and neuroinformatics standards and infrastructure.

The International Neuroinformatics Coordinating Facility (INCF) is an international organization established to coordinate international neuroinformatics infrastructure, and currently has 17 member countries across North America, Europe, Australia, and Asia. With its international network of national nodes, INCF is well positioned to connect scientists from its member countries with international large-scale brain initiatives to strengthen global collaboration and accelerate discovery in neuroscience. The INCF will play an increasingly important role in establishing and operating scientific programs to develop standards for neuroscience data sharing, analysis, modeling, and simulation. The global computational and informatics infrastructure will enable the integration of neuroscience data and knowledge worldwide, and catalyze insights into brain function in health and disease. MASSIVE participation in the Victorian node of the INCF provides an Australian centralized hardware and software facility and a national focal point for imaging and neuroinformatics expertise.

The HPB and the US-led BRAIN Initiative sit alongside a number of other major grand-challenge scientific endeavors, including mapping the human genome or understanding the fabric of matter and the universe using the CERN Large Hadron Collider or the Square Kilometer Array. These endeavors each produce immense volumes of data and are totally reliant on large-scale data processing to uncover new knowledge. Likewise, neuroscience is increasingly a data and simulation driven science and facilities such as MASSIVE are essential to develop new understandings of the brain.

CONCLUSION

Neuroscience and neuroinformatics is an area of priority for the governments of most research intensive countries. Computational HPC approaches are central to neuroscience and to emerging neuroscience technologies including robotics, intelligent systems, and medical bionics. HPC facilities are essential for any future economy based on knowledge intensive industries. MASSIVE provides an Australian centralized hardware and software facility and a focal point for imaging and neuroinformatics expertise. The development of MASSIVE has been based on best practice in system integration methodologies, frameworks, and architectures. MASSIVE is now driving research in advanced brain imaging MRI, x-ray CT, optical microscopy and increasingly synchrotron x-ray and infrared imaging.

REFERENCES

Amazon. (2013). *High Performance Computing (HPC) on AWS* [*Online*]. Amazon. Available online at: http://aws.amazon.com/hpc-applications/ (Accessed October 12, 2013).

Behrens, T., Berg, H. J., Jbabdi, S., Rushworth, M., and Woolrich, M. (2007). Probabilistic diffusion tractography with multiple fibre orientations: what can we gain? *Neuroimage* 34, 144–155. doi: 10.1016/j.neuroimage.2006.09.018

Bellec, P., Lavoie-Courchesne, S., Dickinson, P., Lerch, J. P., Zijdenbos, A. P., and Evans, A. C. (2012). The pipeline system for Octave and Matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows. *Front. Neuroinform.* 6:7. doi: 10.3389/fninf.2012.00007

Buccigrossi, R., Ellisman, M., Grethe, J., Haselgrove, C., Kennedy, D. N., Martone, M., et al. (2007). "The neuroimaging informatics tools and resources clearinghouse (NITRC)," in *AMIA... Annual Symposium Proceedings/AMIA Symposium. AMIA Symposium*, 1000.

CAVE2. (2013). *Monash CAVE2* [Online]. Available online at: http://www.monash.edu.au/cave2 [Accessed 1/12/2013].

DaRIS. (2013). *DaRIS* [Online]. Available online at: http://nsp.nectar.org.au/wiki-its-r/doku.php?id=datamanagement:daris (Accessed November 16, 2013).

Domínguez, J. F., Egan, G. F., Gray, M. A., Churchyard, A., Chua, P., Stout, J. C., et al. (2013). Multi-modal neuroimaging in premanifest and early Huntington's disease: 18 month longitudinal data from IMAGE-HD. *PLoS ONE* 8:e74131. doi: 10.1371/journal.pone.0074131

Duyn, J. H., Van Gelderen, P., Li, T.-Q., De Zwart, J. A., Koretsky, A. P., and Fukunaga, M. (2007). High-field MRI of brain cortical substructure based on signal phase. *Proc. Natl. Acad. Sci. U.S.A.* 104, 11796–11801. doi: 10.1073/pnas.0610821104

Egan, G. F., Barnes, D. G., Killeen, N., Lohrey, J., Liu, W., Goscinksi, W., et al. (2012). "A multi-modality neuroimaging research data informatics system," in *5th International Conference on Neuroinformatic* (Munich).

Ekanayake, J., and Fox, G. (2010). "High performance parallel computing with clouds and cloud technologies," in *Cloud Computing*, eds D. Avresky, M. Diaz, A. Bode, B. Ciciani, and E. Dekel (Berlin, Heidelberg: Springer), 20–38. doi: 10.1007/978-3-642-12636-9_2

Foster, I., and Kesselman, C. (2003). *The Grid 2: Blueprint for a New Computing Infrastructure*. San Francisco, CA: Elsevier.

Frisoni, G. B., Redolfi, A., Manset, D., Rousseau, M.-É., Toga, A., and Evans, A. C. (2011). Virtual imaging laboratories for marker discovery in neurodegenerative diseases. *Nat. Rev. Neurol.* 7, 429–438. doi: 10.1038/nrneurol.2011.99

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J. P., Frith, C. D., and Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Hum. Brain Mapp.* 2, 189–210. doi: 10.1002/hbm.460020402

Georgiou-Karistianis, N., Gray, M. A., Domínguez D, J. F., Dymowski, A. R., Bohanna, I., Johnston, L. A., et al. (2013). Automated differentiation of pre-diagnosis Huntington's disease from healthy control individuals based on quadratic discriminant analysis of the basal ganglia: the IMAGE-HD study. *Neurobiol. Dis.* 51, 82–92. doi: 10.1016/j.nbd.2012.10.001

Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., et al. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome Res.* 15, 1451–1455. doi: 10.1101/gr.4086505

Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., et al. (2011). Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. *Front. Neuroinform.* 5:13. doi: 10.3389/fninf.2011.00013

Goscinski, W. (2013). "Informatics infrastructure for the australian neuroscience community: the multi-modal australian sciences imaging and visualisation environment and the characterisation virtual laboratory," in *Front. Neuroinform. Conference Abstract: Neuroinformatics 2013* (Stockholm: Frontiers).

Goscinski, W., and Gureyev, T. (2011). "The multi-modal australian sciences imaging and visualisation environment (MASSIVE) for near realtime CT reconstruction using XLI," in *eResearch Australasia Conference* (Melbourne).

Gray, M. A., Egan, G. F., Ando, A., Churchyard, A., Chua, P., Stout, J. C., et al. (2013). Prefrontal activity in Huntington's disease reflects cognitive and neuropsychiatric disturbances: the IMAGE-HD study. *Exp. Neurol.* 239, 218–228. doi: 10.1016/j.expneurol.2012.10.020

Grethe, J. S., Baru, C., Gupta, A., James, M., Ludaescher, B., Martone, M. E., et al. (2005). Biomedical informatics research network: building a national collaboratory to hasten the derivation of new understanding and treatment of disease. *Stud. Health Technol. Inform.* 112, 100–110.

Greve, D., Mueller, B., Brown, G., Liu, T., and Glover, G. F. (2010). "Processing methods to reduce intersite variability in fMRI," in *Proceedings* *of the 16th Annual Meeting of the Organization for Human Brain Mapping* (Barcelona).

Gureyev, T. E., Nesterets, Y., Ternovski, D., Thompson, D., Wilkins, S. W., Stevenson, A. W., et al. (2011). "Toolbox for advanced X-ray image processing," in *Proc. SPIE 8141 B, 81410, 81410B-14* (San Diego, CA). doi: 10.1117/12.893252

Haitas, N., and Glatard, T. (2012). "Distributed computing for neurosciences: the N4U example," in *Journées Scientifiques Mésocentres et France Grilles*.

HBP. (2012). *The Human Brain Project: A Report to the European Commission*. Lausanne: The HBP-PS Consortium.

Henderson, A., Ahrens, J., and Law, C. (2004). *The ParaView Guide, 3rd Edn*. Clifton Park, NY: Kitware, Inc. ISBN-10: 1930934211; ISBN-13: 978-1930934214

Janke, A. (2013). *Successful Models, How to Make and Distribute Them, Lessons From The Past and Future Directions*. San Diego, CA: Society For Neuroscience.

Killeen, N. E. B., Lohrey, J. M., Farrell, M., Liu, W., Garic, S., Abramson, D., et al. (2012). "Integration of modern data management practice with scientific workflows," in *Proceedings of 8th IEEE International Conference on eScience* (Chicago).

Laguitton, S., Riviere, D., Vincent, T., Fischer, C., Geffroy, D., Souedet, N., et al. (2011). "Soma-workflow: a unified and simple interface to parallel computing resources," in *MICCAI Workshop on High Performance and Distributed Computing for Medical Imaging*, Toronto.

Lee, J., Shmueli, K., Kang, B. T., Yao, B., Fukunaga, M., Van Gelderen, P., et al. (2012). The contribution of myelin to magnetic susceptibility-weighted contrasts in high-field MRI of the brain. *Neuroimage* 59, 3967–3975. doi: 10.1016/j.neuroimage.2011.10.076

Liu, T., Liu, J., De Rochefort, L., Spincemaille, P., Khalidov, I., Ledoux, J. R., et al. (2011). Morphology enabled dipole inversion (MEDI) from a single-angle acquisition: comparison with COSMOS in human brain imaging. *Magn. Reson. Med.* 66, 777–783. doi: 10.1002/mrm.22816

Liu, T., Spincemaille, P., De Rochefort, L., Kressler, B., and Wang, Y. (2009). Calculation of susceptibility through multiple orientation sampling (COSMOS): a method for conditioning the inverse problem from measured magnetic field map to susceptibility source image in MRI. *Magn. Reson. Med.* 61, 196–204. doi: 10.1002/mrm.21828

Lohrey, J. M., Killeen, N. E., and Egan, G. F. (2009). An integrated object model and method framework for subject-centric e-Research applications. *Front. Neuroinform.* 3:19. doi: 10.3389/neuro.11.019.2009

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., et al. (2006). Scientific workflow management and the Kepler system. *Concurr. Comput.* 18, 1039–1065. doi: 10.1002/cpe.994

Marcus, D. S., Olsen, T. R., Ramaratnam, M., and Buckner, R. L. (2007). The extensible neuroimaging archive toolkit. *Neuroinformatics* 5, 11–33. doi: 10.1385/NI:5:1:11

Markram, H. (2006). The blue brain project. *Nat. Rev. Neurosci.* 7, 153–160. doi: 10.1038/nrn1848

Markram, H. (2012). A countdown to a digital simulation of every last neuron in the human brain. *Sci. Am.* 306.

NeCTAR. (2013). *National eResearch Collaboration Tools and Resources (NeCTAR) Project [Online]*. Available online at: http://www.nectar.org.au/

Nelson, M. R., Reid, C. M., Ames, D. A., Beilin, L. J., Donnan, G. A., Gibbs, P., et al. (2008). Feasibility of conducting a primary prevention trial of low-dose aspirin for major adverse cardiovascular events in older people in Australia: results from the ASPirin in reducing events in the Elderly (ASPREE) pilot study–Research. *Med. J. Aust.* 189, 105–109.

Ng, A. (2013). *Diffusion-Guided Quantitative Susceptibility Mapping*. Salt Lake City, UT: ISMRM.

NITRC. (2013). *NITRC Computational Environment* [Online]. AWS Marketplace. Available online at: https://aws.amazon.com/marketplace/pp/B00AW0MBLO (Accessed October 31, 2013).

OpenStack. (2013). *Openstack Cloud Software* [Online]. Available online at: http://www.openstack.org/ (Accessed October 8, 2013).

Poudel, G. R., Egan, G. F., Churchyard, A., Chua, P., Stout, J. C., and Georgiou-Karistianis, N. (2013). Abnormal synchrony of resting state networks in premanifest and symptomatic Huntington disease: the IMAGE-HD study. *J. Psychiatry Neurosci.* 38, 120226–120226. doi: 10.1503/jpn.120226

Redolfi, A., McClatchey, R., Anjum, A., Zijdenbos, A., Manset, D., Barkhof, F., et al. (2009). Grid infrastructures for computational neuroscience: the neuGRID example. *Future Neurol.* 4, 703–722. doi: 10.2217/fnl.09.53

Rex, D. E., Ma, J. Q., and Toga, A. W. (2003). The LONI pipeline processing environment. *Neuroimage* 19, 1033–1048. doi: 10.1016/S1053-8119(03)00185-X

Richards, K., Watson, C., Buckley, R. F., Kurniawan, N. D., Yang, Z., Keller, M. D., et al. (2011). Segmentation of the mouse hippocampal formation in magnetic resonance images. *Neuroimage* 58, 732–740. doi: 10.1016/j.neuroimage.2011.06.025

Rivers, M. L., and Wang, Y. (2006). "Recent developments in microtomography at GeoSoilEnviroCARS," in *Optics and Photonics*, Vol. 63180J-63180J-15 (San Diego, CA: International Society for Optics and Photonics). doi: 10.1117/12.681144

Schmuck, F. B., and Haskin, R. L. (2002). "GPFS: a shared-disk file system for large computing clusters," in *FAST 02 Proceedings of the 1st USENIX Conference on File and Storage Technologies* (Monterey, CA), 19. doi: 10.1090/S0002-9947-02-03021-0

Sivagnanam, S., Astakhov, V., Yoshimoto, K., Carnevale, T., Martone, M., Majumdar, A., et al. (2013). "A neuroscience gateway: software and implementation," in *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery* (San Diego, CA: ACM), 31.

Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E., Johansen-Berg, H., et al. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage* 23, S208–S219. doi: 10.1016/j.neuroimage.2004.07.051

Ullmann, J. F., Keller, M. D., Watson, C., Janke, A. L., Kurniawan, N. D., Yang, Z., et al. (2012). Segmentation of the C57BL/6J mouse cerebellum in magnetic resonance images. *Neuroimage* 62, 1408–1414. doi: 10.1016/j.neuroimage.2012.05.061

Zalesky, A., Fornito, A., and Bullmore, E. T. (2010). Network-based statistic: identifying differences in brain networks. *Neuroimage* 53, 1197–1207. doi: 10.1016/j.neuroimage.2010.06.041

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 19 December 2013; accepted: 10 March 2014; published online: 27 March 2014.

Citation: Goscinski WJ, McIntosh P, Felzmann U, Maksimenko A, Hall CJ, Gureyev T, Thompson D, Janke A, Galloway G, Killeen NEB, Raniga P, Kaluza O, Ng A, Poudel G, Barnes DG, Nguyen T, Bonnington P and Egan GF (2014) The multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) high performance computing infrastructure: applications in neuroscience and neuroinformatics research. Front. Neuroinform. ***8****:30. doi: 10.3389/fninf.2014.00030*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 21 May 2014
doi: 10.3389/fninf.2014.00052

A simple tool for neuroimaging data sharing

Christian Haselgrove*, Jean-Baptiste Poline and David N. Kennedy

University of Massachusetts Medical School, Worcester, MA, USA

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Maarten Mennes, Radboud University Nijmegen Medical Centre, Netherlands
Zeeshan Ahmed, University of Wuerzburg, Germany

****Correspondence:***
Christian Haselgrove, University of Massachusetts Medical School, 55 Lake Avenue North, Worcester, MA 01655, USA
e-mail: christian.haselgrove@umassmed.edu

Data sharing is becoming increasingly common, but despite encouragement and facilitation by funding agencies, journals, and some research efforts, most neuroimaging data acquired today is still not shared due to political, financial, social, and technical barriers to sharing data that remain. In particular, technical solutions are few for researchers that are not a part of larger efforts with dedicated sharing infrastructures, and social barriers such as the time commitment required to share can keep data from becoming publicly available. We present a system for sharing neuroimaging data, designed to be simple to use and to provide benefit to the data provider. The system consists of a server at the International Neuroinformatics Coordinating Facility (INCF) and user tools for uploading data to the server. The primary design principle for the user tools is ease of use: the user identifies a directory containing Digital Imaging and Communications in Medicine (DICOM) data, provides their INCF Portal authentication, and provides identifiers for the subject and imaging session. The user tool anonymizes the data and sends it to the server. The server then runs quality control routines on the data, and the data and the quality control reports are made public. The user retains control of the data and may change the sharing policy as they need. The result is that in a few minutes of the user's time, DICOM data can be anonymized and made publicly available, and an initial quality control assessment can be performed on the data. The system is currently functional, and user tools and access to the public image database are available at http://xnat.incf.org/.

Keywords: neuroinformatics, neuroimaging, quality assessment, data processing, data archiving

INTRODUCTION

Data sharing is becoming increasingly common (Biswal et al., 2010; Di Martino et al., 2013), but despite encouragement and facilitation by funding agencies, journals, and some labs and larger research efforts[1] (Hall et al., 2012; Prior et al., 2013), there remain political, financial, social, and technical barriers to sharing data (Poline et al., 2012). Excuses such as "it's too hard" and "it takes too long" are all too common, and there is anxiety about subject protection and control of data (De Schutter, 2010). And unless one is part of a large project with dedicated sharing infrastructure, there is also a lack of open technical infrastructure and public and free archive space.

There are some central, open databases for image data sharing such as The Cancer Imaging Archive[2] and the National Database for Autism Research[3], but these are domain-specific, and contributing data requires a substantial investment of time to handle both bureaucratic and technical aspects of contributing data. On the other end of the spectrum are image databases that can be installed locally, such as COINS[4] (Scott et al., 2011), the Human Imaging Database[5] (Ozyurt et al., 2010), LORIS[6] (Das et al., 2012), NIDB[7] (Book et al., 2013), and XNAT[8] (Marcus et al., 2007). Using any of these to share image data requires an investment in hardware as well as initial and ongoing technical support. With the exception of XNAT Central, none of these provide a public, open instance that anyone can use to share their data.

Given an open repository such as XNAT Central, other issues come into play. The actual mechanics of uploading data must then be addressed. There are tools available to facilitate data upload, but these often require somewhat involved installation, and most are then general in scope, with many options that must be understood. XNAT Desktop[9] and DicomBrowser[10], for instance, allow a user to manage local data and send it to XNAT Central, but the flexibility in anonymization options and subject identifier customization mean that there is a learning curve to using these tools effectively. Moreover, they often don't capture the relevant metadata simply and efficiently.

We have created a system for data sharing that attempts to address many of these issues. We set up a public, open image repository within an international organization that can host and manage imaging data, and have created user tools that make data upload to this server trivial. The user software is designed to be easy to install, and once installed, data upload is initiated

[1] http://grants.nih.gov/grants/policy/data_sharing

[2] http://www.cancerimagingarchive.net/

[3] http://ndar.nih.gov/

[4] http://coins.mrn.org/

[5] http://www.nitrc.org/projects/hid/

[6] https://www.nitrc.org/projects/loris/

[7] http://nidb.sourceforge.net/

[8] http://www.xnat.org

[9] http://www.wiki.xnat.org/display/XNAT/XNAT+Desktop

[10] http://nrg.wustl.edu/software/dicom-browser/

by a simple drag and drop. The user is then walked through the few steps necessary to anonymize and upload the data in a way that control of the data on the repository is retained. On receipt of the data, the repository also runs quality assessment (QA) routines on the data as a service to the user and as additional motivation to share. In the near future, this should also provide the imaging community with a useful resource for quality checking. This report describes the design and implementation of this system and initial results of its testing and validation.

METHODS

OVERVIEW

The system has two components: the image repository and the user tools. The repository is an XNAT installation, and while some XNAT customizations were necessary, most of the innovation lies with the user tools. An overview of the design of the system can be found in **Figure 1**. Since the overarching goal of this system is to make data sharing simple, we describe the components of the system in the order they are encountered by the data as it moves from a local disk to the server. The ultimate effect is that given a few minutes of a researcher's attention, data is anonymized, archived, and shared, and the researcher gets feedback on the quality of the data.

The server itself can be found at http://xnat.incf.org, and the user tools can be downloaded from this location as well. Source code for the user tools and custom code for the repository can be found on GitHub at http://github.com/incf/one_click.

USER UPLOAD TOOL

The driving design principle for the user tool is ease of use. Our goal is to remove the barriers to data sharing, and the more difficult it is to install or successfully run any tool, the more likely it is that the user will give up. We provide two user tools, a command line script and a graphical user interface (GUI). The two options provide the same functionality, but in different ways: the command line script is useful for users comfortable at the command line, while the GUI uploader is useful for users accustomed to a more interactive experience. The only requirement for these tools is that data is prepared in a certain well-defined way before being sent to the archive (see below).

The current user tools are written in Python, released under the BSD license, and can be installed on Linux or Mac OS machines. Dependencies are pydicom[11], httplib2[12], and DCMTK[13]. The user tools can be downloaded directly from the International Neuroinformatics Coordinating Facility (INCF) web site[14]. The command line tool requires manual installation of the dependencies, although it is packaged and released through NeuroDebian[15] (Halchenko and Hanke, 2012) which simplifies installation and dependency handling on Debian systems. The Linux GUI tool also requires PyQt[16]. All of the dependencies are bundled for the Mac OS GUI.

The custom code for the archive server is also made available on line via GitHub[17] and released under the BSD license. Although we plan to support the ability to push to alternate archives, focus so far has been on the user tools and user experience, with one archive sufficient for testing. Similar to a new user tool, a new archive for this system would only have to conform to certain well-defined specifications, such as being able to handle data prepared as described below.

Data selection, validation, and annotation

The first step is selecting the Digital Imaging and Communications in Medicine (DICOM) data to share. This can be invocation of the command line script that takes the containing directories of the data as arguments or dragging and dropping a folder containing data onto the GUI tool (**Figure 2**). The selected data is then validated and sorted into subjects and imaging sessions: the user tool scans the specified directories for DICOM data using pydicom and groups the data by subject (by the DICOM Patient ID field) and imaging session (by Study Instance UID).

If valid data is found, the user is asked to consent to a simple usage agreement before proceeding (**Figure 3**). This agreement is intentionally broad and simple; waiting to implement this upload system until all of the legal aspects of sharing have been perfected is a recipe for failure. The user is then prompted for a user name and password that identify the user on the INCF portal[18]. The

[11] http://code.google.com/p/pydicom/

[12] http://code.google.com/p/httplib2/

[13] http://dicom.offis.de/dcmtk.php.en

[14] http://xnat.incf.org/

[15] http://neuro.debian.net/

[16] http://www.riverbankcomputing.com/software/pyqt/intro

[17] http://github.com/incf/one_click

[18] http://www.incf.org/

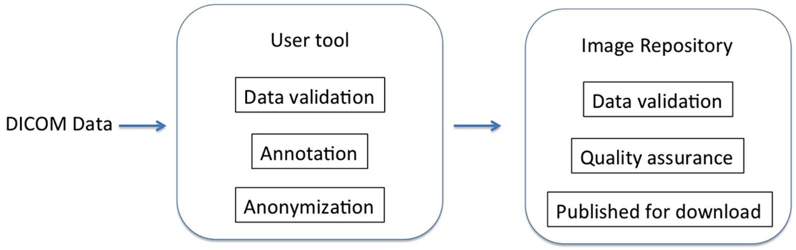


FIGURE 1 | System overview. Users are walked through data preparation using the user tool, after which the data is sent to the image repository for further processing and publishing.

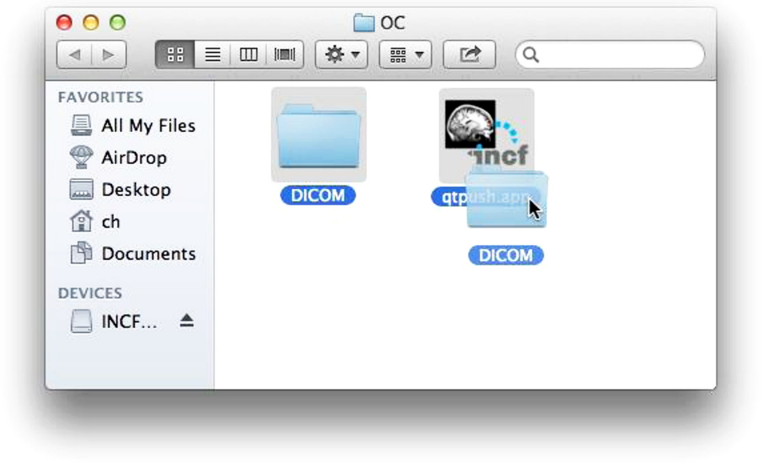
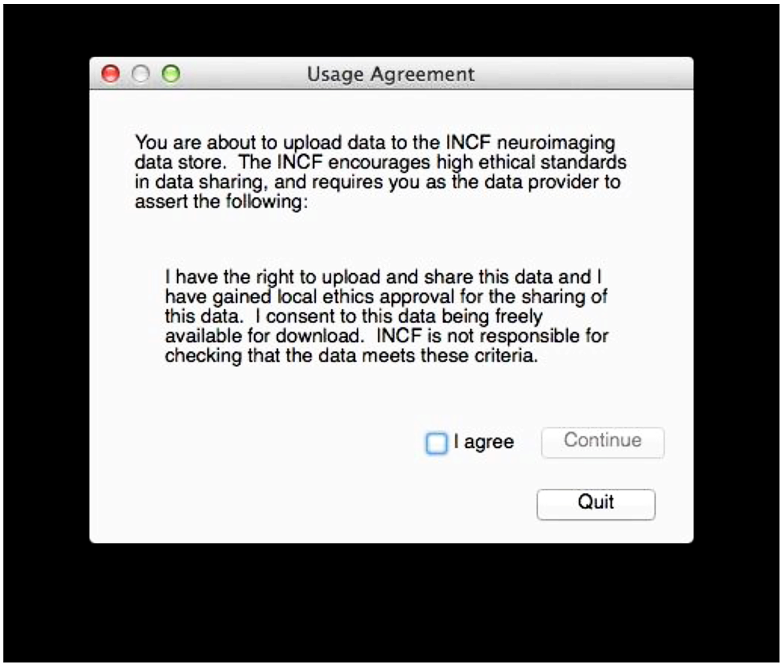


FIGURE 2 | Selecting data for upload. Dragging and dropping the DICOM folder to the uploader application initiates the process using the GUI user tool.

user name allows the archive to assign the data to the user so they retain control of the data, and links to the e-mail address to which reports are sent. Since the archive shares the users and passwords of the INCF portal, the password allows the user tool to query the archive for existing data under the user's control to avoid collisions of new subject or session identifiers (**Figure 4**). This all takes a few short minutes of the user's time and attention.

There is some coordination with the archive required at this stage. The archive server is running XNAT, which provides a set of REST[19] services that allow these queries. XNAT structures data hierarchically into projects, subjects, and sessions. Permissions are handled at the project level: access to subjects and sessions depend solely on the level of access permitted to the containing project. The user tool prompts the user for a project for each subject in the selected data, and since the tool has queried the archive, the

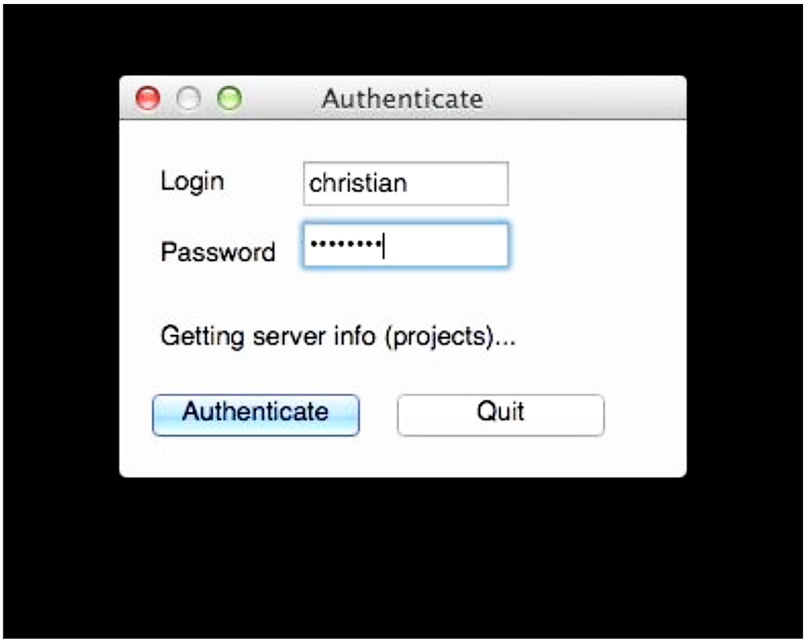
[19] Representational state transfer, an architectural standard for communication between components in a distributed system.



You are about to upload data to the INCF neuroimaging data store. The INCF encourages high ethical standards in data sharing, and requires you as the data provider to assert the following:

I have the right to upload and share this data and I have gained local ethics approval for the sharing of this data. I consent to this data being freely available for download. INCF is not responsible for checking that the data meets these criteria.

FIGURE 3 | Upload agreement. After verifying that DICOM data is available in the selected folder and before further action, the user must consent to this agreement.



Authenticate

Login christian

Password ••••••••

Getting server info (projects)...

FIGURE 4 | User authentication. The uploaded data is tagged with the user name so the user retains control of the data. Requiring the password at this stage allows the tool to query the archive for existing data so conflicts can be avoided when labeling the data. Here, the user tool is querying the archive for existing projects.

tool can verify that the user is specifying a project to which he has access or a new project that can be created. Similarly, the user tool will prompt the user for valid subject and session identifiers that do not conflict with those already in the archive (**Figure 5**).

Anonymization and upload

The data is anonymized locally before it leaves the user's computer and is then sent to the archive. At this stage, all necessary information has been collected from the user, and the data must be prepared and sent to the archive. One benefit of using DICOM data in our initial test case is that the DICOM standard includes a network communication protocol for transferring data, a protocol which XNAT handles natively on the receiving end. But the user tool must first anonymize the data so no identifiable information leaves the user's machine and then annotate the data with the user information and the specified data identifiers. Depending on the amount of data and the quality of the network connection, this may take an hour or more, but it does not require the user's attention.

Anonymization is a challenge because of the various levels and interpretation of anonymization that can be applied. DICOM defines concepts such as patient name and study date that it stores in fields, and there are several different conflicting DICOM anonymization schemes that specify what information should be protected (meaning, in our case, removed from the data). We can illustrate this challenge by examining three different examples of existing anonymization protocols: DICOM Supplement 55[20] (developed primarily with clinical uses in mind), the National Cancer Institute deidentification profile[21], and the default deidentification profile provided by XNAT's DICOM Browser[22]. All agree that the Patient's Name field should be protected, but only one specifies protecting Study Date, another protects Patient's Address, one pair protects Patient's Age, and another pair protects Institution Name, and so on in every combination. Clearly, no consensus is to be found: the level of anonymization depends on the application context and the specifics of the data. In addition, the DICOM specification defines fields that must be present in valid data sets[23], and programs at both the sending and receiving ends of the network transfer have their own quirks regarding what fields they require to be present.

Rather than trying to definitively solve this problem, we decided to choose a set of protected fields that are removed or replaced (guided by existing anonymization profiles), making sure that the network tools on either end would function with our anonymized data. **Table 1** shows the protected fields that are currently removed from the data before it is sent to the archive.

The INCF user name and the project, subject, and session identifiers specified by the user are stored in the Study Comments field, which is replaced or created as needed.

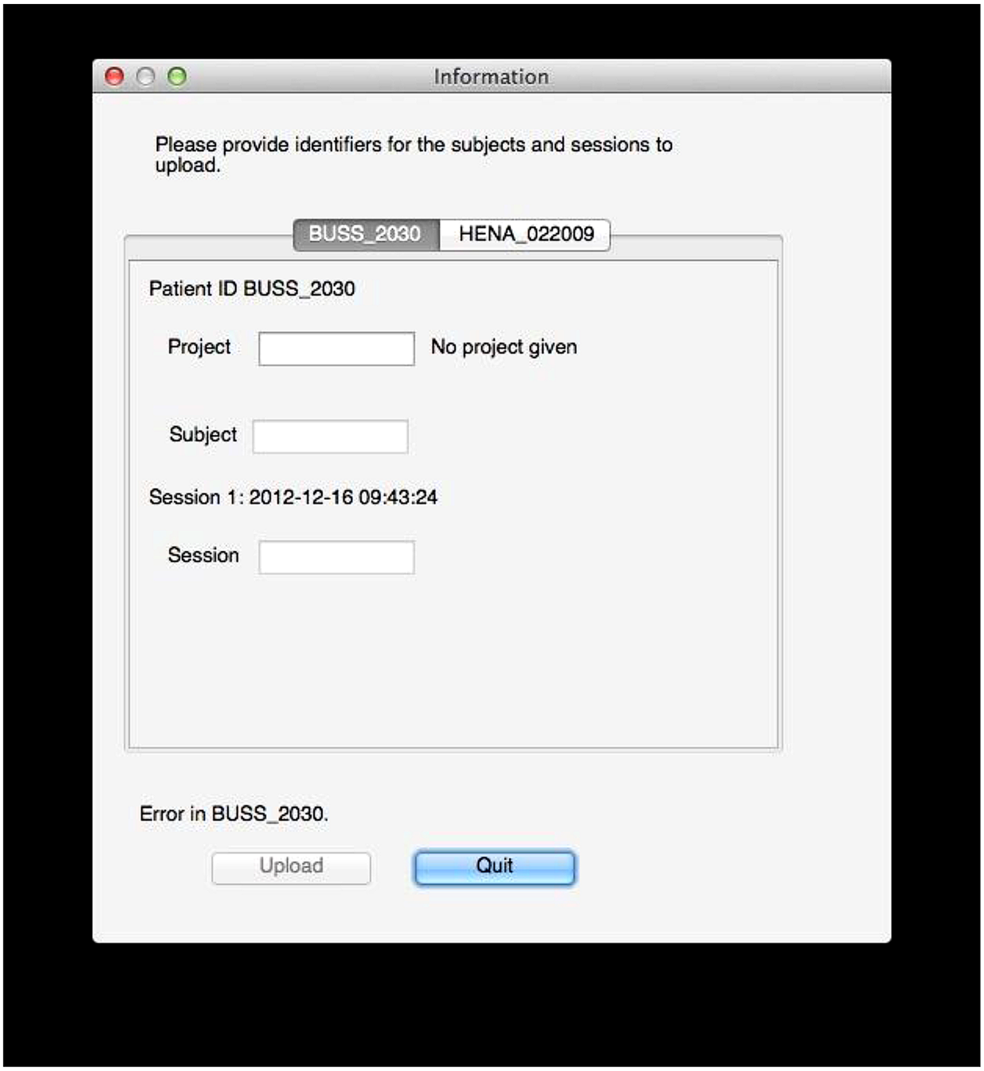
This anonymized and annotated DICOM data is then pushed to the archive using the DICOM network transport protocol by storescu from the DCMTK package. Similar to the Python dependencies described above, this can be installed separately, but

[20] ftp://medical.nema.org/medical/dicom/final/sup55_ft.pdf

[21] https://wiki.nci.nih.gov/display/Imaging/Clinical+Imaging+Data+Sharing

[22] http://nrg.wustl.edu/software/dicom-browser/

[23] http://medical.nema.org/



Please provide identifiers for the subjects and sessions to upload.

Patient ID BUSS_2030

Project No project given

Subject

Session 1: 2012-12-16 09:43:24

Session

Error in BUSS_2030.

FIGURE 5 | Data labeling. In this example, the user tool found data for two subjects, BUSS_2030 and HENA_022009. The user now selects a project and specifies public subject and session identifiers for the data on the archive. Validation is done on the fly: here, an error exists because no project is given, but the tool will also inform the user if he does not have sufficient permissions to upload to the specified project, if the session already exists, if identifiers use invalid characters, and so on.

NeuroDebian handles its installation on Linux and it is bundled with the Mac OS GUI tool.

IMAGE REPOSITORY

The repository itself is located at and hosted by the International Neuroinformatics Coordinating Facility (INCF). The server itself is a Linux virtual machine with two 2.4 GHz processors and a total of 4 GB memory. The image repository is a customized installation of XNAT 1.5.4.

Data validation and archiving

Data is validated on arrival at the archive and then archived. The server itself does not have the processing power or memory for intensive parallel analysis, so launching this computationally intensive processing immediately when data arrives could easily overload the system if a lot of data arrives at once. This step is therefore queued and run using the arc-queue tools[24].

The validation processing starts with an anonymization check, and if the data does not conform to the anonymization profile described above (i.e., if any of the protected fields are found in the data), the data is removed from the archive and the user is notified by e-mail. The content of the Study Comments field is then validated, checking for a valid user and for project, subject, and session identifiers. If the project exists, user permissions are also checked. If everything is in order, archiving begins.

The archiving itself is a standard, built-in function of XNAT, which arranges the data into projects, subjects, sessions, and scans, after which thumbnail images are created for each scan (**Figure 6**).

[24] http://www.nitrc.org/projects/xnat_extras/

Table 1 | DICOM fields for anonymization.

Tag	Name
(0008, 0050)	Accession number
(0008, 0080)	Institution name
(0008, 0090)	Referring physician's name
(0008, 0096)	Referring physician identification
(0008, 1048)	Physician(s) of record
(0008, 1049)	Physician(s) of record identification
(0008, 1050)	Performing physicians' name
(0008, 1052)	Performing physician identification
(0008, 1060)	Name of physician(s) reading study
(0008, 1062)	Physician(s) reading study identification
(0010, 0030)	Patient's birth date
(0010, 0050)	Patient's insurance plan code
(0010, 0101)	Patient's primary language code
(0010, 1000)	Other patient IDs
(0010, 1001)	Other patient names
(0010, 1002)	Other patient IDs
(0010, 1005)	Patient's birth name
(0010, 1010)	Patient's age
(0010, 1040)	Patient's address
(0010, 1060)	Patient's mother's birth name

The current user tools clear or remove values for these fields, and data that arrives at the archive with any of these fields set is rejected.

At this point the data is available for download, and users can browse or search the archive for data. After archiving, QA is launched.

Quality assessment

Once the data has been validated, QA runs are launched. QA procedures differ for various scan types, and the results are stored on the archive and sent to the user by e-mail. Currently, three types of QA are available for these scan types: structural, time series, and diffusion. Structural QA is run on any scan of type MPRAGE. Time series and diffusion QA is launched for every scan and allowed to fail if the data does not satisfy the prerequisites for these types (i.e., data with only one time point will file the time series QA, and data without diffusion gradient direction descriptions will fail the diffusion QA). QA begins by converting each scan to NIfTI-1 and NRRD, and the bundling the data and descriptors into an XCEDE-formatted file (Gadde et al., 2012). XCEDE-formatted data is required by the QA procedures. Even if the QA fails, these alternate data formats will be available for download on the archive.

Structural QA. The structural QA is a custom procedure created for this system. This procedure calculates image intensity statistics over white matter, gray matter, CSF, whole brain, and the region exterior to the head. The signal to noise ratio (SNR) is defined as the mean image intensity in the brain divided by the standard deviation of the image intensity external to the brain.

FSL[25] (Zhang et al., 2001; Smith, 2002; Smith et al., 2004; Jenkinson et al., 2005) is used to classify regions in the volume and calculate statistics, specifically:

- Brain and head are determined using bet image -A -m.
- Tissue types are determined using fast -t 1 image_brain, where image_brain is an output of bet.
- Statistics are calculated using fslstats, using -k to mask each region, -R for the minimum and maximum intensities, -r for the robust minimum and maximum intensities, -m for the mean intensity, -s for the intensity standard deviation, -v for the number of voxels and the volume.

As this is a new and custom structural image QA procedure designed as a simple proof of concept for this tool, it is imperfect and likely to evolve as it is used as we study the results obtained on large numbers of scans.

Time series QA. Time series QA is performed by fmriqa_generate.pl, part of the BXH/XCEDE Tools suite[26] (Friedman et al., 2006). This program takes XCEDE wrapped data and produces a web page reporting the results, including several plots. Examples of measures are the mean volume intensity at each time point and the center of mass (x, y, and z) at each time point. Plots of these measures can indicate at a glance if there is a variation at a given time point that warrants further investigation. The mean SNR and mean signal to fluctuation noise ratio (SFNR) are also calculated as part of this process.

Diffusion QA. Diffusion QA is provided by DTIPrep[27] (Liu et al., 2010) with default parameters (DTIPrep -w scan.nrrd -p default -d -c). The DTIPrep produces an XML report containing a number of pass/fail checks of basic image parameters (spatial information, basic gradient checks) followed by informational reports of other parameters (e.g., gradient directions) that can be examined for errors or possible problems. DTIPrep will also generate warnings of certain non-standard conditions that might warrant additional investigation (e.g., a non-standard number of gradient directions or suspicious b-values).

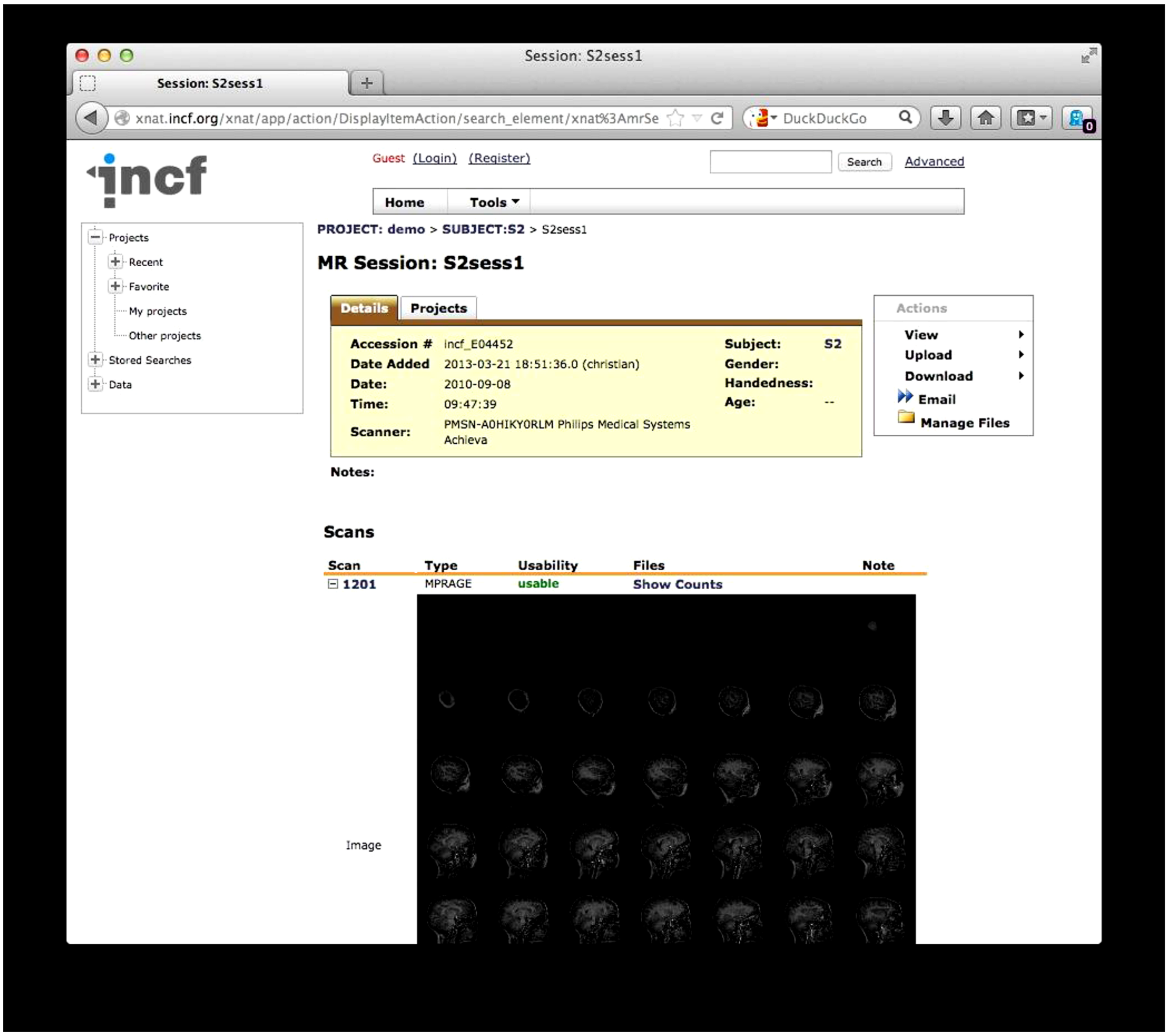
QA reporting. Quality assessment results are parsed and stored on the archive as assessments, custom XNAT data types that allow for storage, management, and display of arbitrary data types. These assessments are accessible from the web front-end and are associated with the raw data for each scan (**Figure 7**).

While the diffusion QA is mainly informational with some pass/fail results, the SNR and SFNR calculated for the structural and time series QA procedures provide quantitative values that may not have much meaning in isolation but can be compared against other scans or collections of scans. For these QA reports, histograms of SNR and SFNR for similar scans in the database as

[25] http://www.fmrib.ox.ac.uk/fsl/

[26] http://www.nitrc.org/projects/bxh_xcede_tools

[27] http://www.nitrc.org/projects/dtiprep



PROJECT: demo > SUBJECT:S2 > S2sess1

MR Session: S2sess1

Accession # incf_E04452
Date Added 2013-03-21 18:51:36.0 (christian)
Date: 2010-09-08
Time: 09:47:39
Scanner: PMSN-A0HIKYORLM Philips Medical Systems Achieva

Subject: S2
Gender:
Handedness:
Age: --

Notes:

Scans

Scan	Type	Usability	Files	Note
⊟ 1201	MPRAGE	usable	Show Counts	

Image

FIGURE 6 | The existing one-click XNAT archive. Data is structured by project, subject, session, and scan. An automatically generated thumbnail image is also shown.

well as for data in the 1000 Functional Connectomes[28], as a reference dataset, are generated on the fly to give context of the SNR and SFNR values for these scans.

The archive web front-end presents other data as is, such as the raw values of tissue volume and voxel intensity statistics for each tissue type (structural QA), the intensity and motion plots (time series QA), and the diffusion pass/fail checks and gradient information (diffusion QA).

When this processing is complete, the user is notified by e-mail and given pointers to the data and to the QA results.

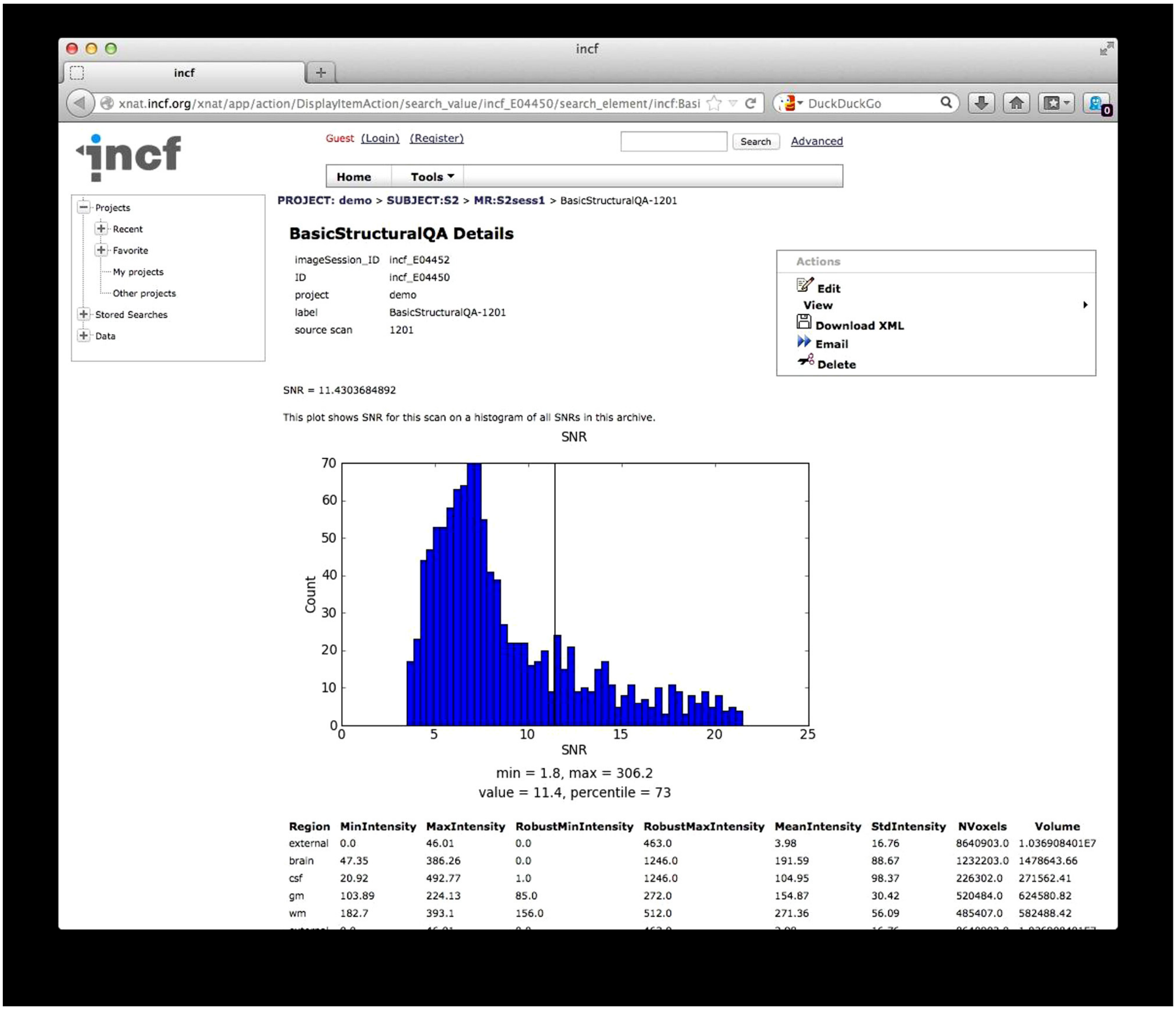
Data sharing

The data itself and the QA results are archived in a structured way and made publicly available in several formats. The user retains full control of the data, however, and can make the data private (on a project-by-project basis, following the XNAT security model) or can remove the data from the archive completely.

DISCUSSION

The system was conceived to remove some of the technical barriers to data sharing and address some common excuses such as "it's too hard," "it takes too long," "there's nowhere that will publicly host my data," and "I need to make sure the data is anonymized." At this point, the system addresses all of these issues. With this basic functionality in place, the system can support other missions as well. There has been interest in this platform to support the NIH data sharing mandate and journals' data sharing requirements. There has also been independent interest in QA measures and interest in the system providing further basic data analysis such as

[28] http://www.nitrc.org/projects/fcon_1000



PROJECT: demo > SUBJECT:S2 > MR:S2sess1 > BasicStructuralQA-1201

BasicStructuralQA Details

imageSession_ID	incf_E04452
ID	incf_E04450
project	demo
label	BasicStructuralQA-1201
source scan	1201

SNR = 11.4303684892

This plot shows SNR for this scan on a histogram of all SNRs in this archive.

min = 1.8, max = 306.2
value = 11.4, percentile = 73

Region	MinIntensity	MaxIntensity	RobustMinIntensity	RobustMaxIntensity	MeanIntensity	StdIntensity	NVoxels	Volume
external	0.0	46.01	0.0	463.0	3.98	16.76	8640903.0	1.036908401E7
brain	47.35	386.26	0.0	1246.0	191.59	88.67	1232203.0	1478643.66
csf	20.92	492.77	1.0	1246.0	104.95	98.37	226302.0	271562.41
gm	103.89	224.13	85.0	272.0	154.87	30.42	520484.0	624580.82
wm	182.7	393.1	156.0	512.0	271.36	56.09	485407.0	582488.42

FIGURE 7 | Quality Assessment (QA) results. The results from the structural QA are shown as a custom XNAT assessment. The SNR is plotted against a histogram of SNR values for a base set of data and for data in the archive. Raw values from the structural QA are shown at the bottom. Similar reports are created for time series QA and diffusion QA.

FreeSurfer[29] (Dale et al., 1999; Fischl et al., 1999) reconstructions as a matter of course.

LIMITATIONS

There are various limitations to the current system, on the user side, on the repository side, and on the system as a whole.

At this point, the user tools trade customizability for ease of use, but this does not have to be a strict tradeoff. Anonymization should be flexible, and the target of the upload should be customizable (allowing for multiple archives; this could mean archives with other processing on the back end, or local archives). With sensible defaults in place, adding these options does not need to stand in the way of basic usability. The tools do require user attention, but could be even more useful if a non-interactive mode were provided. The command line script could then be embedded in processing or other pipelines so data can be uploaded to an archive as part of the same mechanism that moves it from the scanner to a local lab for analysis, or to use an archive to do some initial analysis. The user tools are also limited as to what platforms they will run on (for the GUI tools), and the command-line script has several dependencies that must be installed by hand if the Debian package is not used. A web-based option for the user tool would be the ideal solution here, but would require that anonymization be performed on the server side or using local JavaScript code.

On the server side, we identify scans for structural QA by their declared scan types (MPRAGE). This could be extended by using a lexicon of scan types (MPRAGE, SPGR, FSPGR, etc) but

[29] http://freesurfer.net/

this solution will not scale: much structural data will be always described in terms unfamiliar to the system, and the lexicon will be forever chasing data found in the real world. A better way of identifying scan types is likely to by inspection of scan parameters reported in DICOM fields combined with a lexicon of scan types. Allowing the user to specify the scan type unambiguously would also solve this problem.

One limitation of the system as a whole is its requirement for DICOM data. While the DICOM transfer protocol was useful for this initial prototype, other data formats (NIfTI-1, MGH, MINC, etc.) are more prevalent in day-to-day use within individual laboratories, and there is currently no good way to convert these files back to DICOM to prepare it for upload. Most imaging data starts as DICOM at the scanner, however, so this limitation is less of a problem as investigators begin to consider centralized archival of their data immediately upon acquisition. The restriction to DICOM data also limits the system to imaging data, while other modalities (e.g., EEG) are excluded from using the system.

Finally, the utility of the structural QA technique is currently unknown. We hope that as this is applied to more data, it will become clear how to interpret it and how to improve it. While the time series and diffusion QA procedures have been formalized more completely, it still remains to be seen exactly how to incorporate these metrics into practical implementations that indicate QA limits for data as a function of a desired use.

CONCLUSION

What was conceived during a discussion of data sharing as a system to aid data sharing has now been implemented, providing users with a way to share data that addresses ease of use, anonymization, and storage and archiving, and even providing some basic processing results. The basic functionality is in place; users need only to start using the system. The fact that they haven't is not a failure of the system; rather, it is a form of progress in ongoing data sharing efforts.

Providing this system that functions to its technical specifications has removed certain technical barriers, throwing into relief some of the social issues standing in the way of effective data sharing. Exposing these issues will allow us to better understand and focus on them. With "we can't share" out of the way, we can better attack "we won't share." Data sharing has not been solved, but the discussion has been moved forward. And as further barriers are removed, we have in place an infrastructure for sharing and archiving.

ACKNOWLEDGMENTS

The authors thank the members of the INCF Neuroimaging Data Sharing Task Force for their input to this project, in particular Dan Marcus for the initial discussion and ideas. Janis Breeze's support and enthusiasm ensured that these good ideas got the momentum and support necessary to make them a reality. Much of the work was carried out under contracts from the INCF, and we thank the current INCF staff, in particular Linda Lanyon and Mathew Abrams for their great help and support.

REFERENCES

Biswal, B. B., Mennes, M., Zuo, X. N., Gohel, S., Kelly, C., Smith, S. M., et al. (2010). Toward discovery science of human brain function. *Proc. Natl. Acad. Sci. U.S.A.* 107, 4734–4739. doi: 10.1073/pnas.0911855107

Book, G. A., Anderson, B. M., Stevens, M. C., Glahn, D. C., Assaf, M., and Pearlson, G. D. (2013). Neuroinformatics Database (NiDB) – a modular, portable database for the storage, analysis, and sharing of neuroimaging data. *Neuroinformatics* 11, 495–505. doi: 10.1007/s12021-013-9194-1

Dale, A. M., Fischl, B., and Sereno, M. I. (1999). Cortical surface-based analysis. I. Segmentation and surface reconstruction. *Neuroimage* 9, 179–194. doi: 10.1006/nimg.1998.0395

Das, S., Zijdenbos, A. P., Harlap, J., Vins, D., and Evans, A. C. (2012). LORIS: a web-based data management system for multi-center studies. *Front. Neuroinform.* 5:37. doi: 10.3389/fninf.2011.00037

De Schutter, E. (2010). Data publishing and scientific journals: the future of the scientific paper in a world of shared data. *Neuroinformatics* 8, 151–153. doi: 10.1007/s12021-010-9084-8

Di Martino, A., Yan, C. G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., et al. (2013). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Mol. Psychiatry* doi: 10.1038/mp.2013.78 [Epub ahead of print].

Fischl, B., Sereno, M. I., and Dale, A. M. (1999). Cortical surface-based analysis. II: inflation, flattening, and a surface-based coordinate system. *Neuroimage* 9, 195–207. doi: 10.1006/nimg.1998.0396

Friedman, L., Glover, G. H., and Fbirn Consortium. (2006). Reducing interscanner variability of activation in a multicenter fMRI study: controlling for signal-to-fluctuation-noise-ratio (SFNR) differences. *Neuroimage* 33, 471–481. doi: 10.1016/j.neuroimage.2006.07.012

Gadde, S., Aucoin, N., Grethe, J. S., Keator, D. B., Marcus, D. S., Pieper, S., et al. (2012). XCEDE: an extensible schema for biomedical data. *Neuroinformatics* 10, 19–32. doi: 10.1007/s12021-011-9119-9

Halchenko, Y. O., and Hanke, M. (2012). Open is not enough. Let's take the next step: an integrated, community-driven computing platform for neuroscience. *Front. Neuroinform.* 6:22. doi: 10.3389/fninf.2012.00022

Hall, D., Huerta, M. F., McAuliffe, M. J., and Farber, G. K. (2012). Sharing heterogeneous data: the national database for autism research. *Neuroinformatics* 10, 331–339. doi: 10.1007/s12021-012-9151-4

Jenkinson, M., Pechaud, M., and Smith, S. (2005). "BET2: MR-based estimation of brain, skull and scalp surfaces," in *Proceedings of the Eleventh Annual Meeting of the Organization for Human Brain Mapping*, Oxford, UK.

Liu, Z., Want, Y., Gerig, G., Gouttard, S., Tao, R., Fletcher, T., et al. (2010). Quality control of diffusion weighted images. *Proc. Soc. Photo Opt. Instrum. Eng.* 7628. doi: 10.1117/12.844748

Marcus, D. S., Olsen, T. R., Ramaratnam, M., and Buckner, R. L. (2007). The extensible neuroimaging archive toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34. doi: 10.1385/NI:5:1:11

Ozyurt, I. B., Keator, D. B., Wei, D., Fennema-Notestine, C., Pease, K. R., Bockholt, J., et al. (2010). Federated web-accessible clinical data management within an extensible neuroimaging database. *Neuroinformatics* 8, 231–249. doi: 10.1007/s12021-010-9078-6

Poline, J. B., Breeze, J. L., Ghosh, S., Gorgolewski, K., Halchenko, Y. O., Hanke, M., et al. (2012). Data sharing in neuroimaging research. *Front. Neuroinform.* 6:9. doi: 10.3389/fninf.2012.00009

Prior, F. W., Clark, K., Commean, P., Freymann, J., Jaffe, C., Kirby, J., et al. (2013). TCIA: an information resource to enable open science. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2013, 1282–1285. doi: 10.1109/EMBC.2013.6609742

Scott, A., Courtney, W., Wood, D., de la Garza, R., Lane, S., King, M., et al. (2011). COINS: an innovative informatics and neuroimaging tool suite built for large heterogeneous datasets. *Front. Neuroinform.* 5:33. doi: 10.3389/fninf.2011.00033

Smith, S. M. (2002). Fast robust automated brain extraction. *Hum. Brain Mapp.* 17, 143–155. doi: 10.1002/hbm.10062

Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E. J., Johansen-Berg, H., et al. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage* 23, 208–219. doi: 10.1016/j.neuroimage.2004.07.051

Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imaging* 20, 45–57.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 20 December 2013; accepted: 28 April 2014; published online: 21 May 2014.

Citation: Haselgrove C, Poline J-B and Kennedy DN (2014) A simple tool for neuroimaging data sharing. Front. Neuroinform. ***8****:52. doi: 10.3389/fninf.2014.00052*

This article was submitted to the journal Frontiers in Neuroinformatics.

METHODS ARTICLE
published: 05 June 2014
doi: 10.3389/fninf.2014.00060

Automated collection of imaging and phenotypic data to centralized and distributed data repositories

Margaret D. King[1]*, Dylan Wood[1], Brittny Miller[1], Ross Kelly[1], Drew Landis[1], William Courtney[1], Runtang Wang[1], Jessica A. Turner[1,2] and Vince D. Calhoun[1,3]

[1] The Mind Research Network, Albuquerque, NM, USA
[2] Department of Psychology, Georgia State University, Atlanta, GA, USA
[3] Departments of Electrical and Computer Engineering, Neurosciences, Computer Science, and Psychiatry, University of New Mexico, Albuquerque, NM, USA

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Ariana Anderson, University of California, Los Angeles, USA
David A. Gutman, Emory University, USA

***Correspondence:**
Margaret D. King, The Mind Research Network, 1101 Yale Blvd. NE, Albuquerque, NM 87106, USA
e-mail: mking@mrn.org

Accurate data collection at the ground level is vital to the integrity of neuroimaging research. Similarly important is the ability to connect and curate data in order to make it meaningful and sharable with other investigators. Collecting data, especially with several different modalities, can be time consuming and expensive. These issues have driven the development of automated collection of neuroimaging and clinical assessment data within COINS (Collaborative Informatics and Neuroimaging Suite). COINS is an end-to-end data management system. It provides a comprehensive platform for data collection, management, secure storage, and flexible data retrieval (Bockholt et al., 2010; Scott et al., 2011). It was initially developed for the investigators at the Mind Research Network (MRN), but is now available to neuroimaging institutions worldwide. Self Assessment (SA) is an application embedded in the Assessment Manager (ASMT) tool in COINS. It is an innovative tool that allows participants to fill out assessments via the web-based Participant Portal. It eliminates the need for paper collection and data entry by allowing participants to submit their assessments directly to COINS. Instruments (surveys) are created through ASMT and include many unique question types and associated SA features that can be implemented to help the flow of assessment administration. SA provides an instrument queuing system with an easy-to-use drag and drop interface for research staff to set up participants' queues. After a queue has been created for the participant, they can access the Participant Portal via the internet to fill out their assessments. This allows them the flexibility to participate from home, a library, on site, etc. The collected data is stored in a PostgresSQL database at MRN. This data is only accessible by users that have explicit permission to access the data through their COINS user accounts and access to MRN network. This allows for high volume data collection and with minimal user access to PHI (protected health information). An added benefit to using COINS is the ability to collect, store and share imaging data *and* assessment data with no interaction with outside tools or programs. All study data collected (imaging and assessment) is stored and exported with a participant's unique subject identifier so there is no need to keep extra spreadsheets or databases to link and keep track of the data. Data is easily exported from COINS via the Query Builder and study portal tools, which allow fine grained selection of data to be exported into comma separated value file format for easy import into statistical programs. There is a great need for data collection tools that limit human intervention and error while at the same time providing users with intuitive design. COINS aims to be a leader in database solutions for research studies collecting data from several different modalities.

Keywords: assessment data collection, neuroinformatics, tool suite, database, intuitive, COINS

INTRODUCTION

Collecting phenotypic data is a central part of any neuroimaging study. Traditionally, this data has been collected by writing observations and responses on paper. In some cases, study staff will record the data on paper while interviewing the participant. In other cases, the participant may enter the data directly onto the paper themselves. After this initial data collection, the paper hard-copies must be carefully cataloged and stored in filing systems. Since data contained on sheets of paper is difficult to analyze, the data must then be entered into a computer system (e.g., database or spreadsheet). In order to reduce errors, many studies will perform dual entry, where the data is redundantly entered by two individuals. The two entries are then compared, and any differences are resolved before an official entry is created. Even with dual entry, there is a small chance of data entry errors.

Fortunately, modern technology has provided researchers with many alternatives to the expensive, time-consuming process described above. Data collection services like SurveyMonkey[1], Mechanical Turk, and Qualtrics[2] (Buhrmester et al., 2011) offer comprehensive form building tools. Once built, a form can be used by study staff or a participant to enter data directly into a computer, thus avoiding the cost and time associated with entering data on paper records into a computer system. The data collected in using these systems must still be securely tied to each study participant, and their imaging data (typically stored on local databases, with metadata contained in spreadsheets). Managing the connections between electronic phenotypic data and the participant records in a way that does not compromise participant privacy is a stressful and time consuming task.

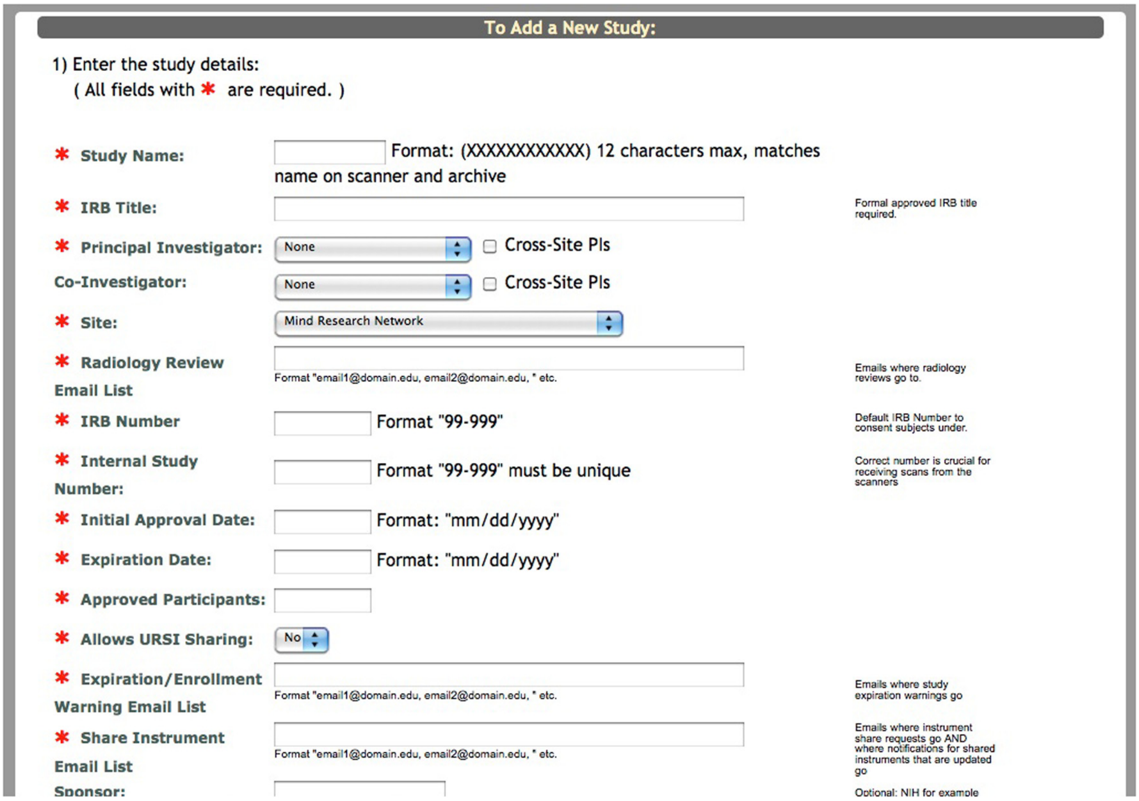
In this article we introduce the web-based Self Assessment tool as an optimal method for assessment data collection. The impetus for developing this tool was to reduce data collection and entry time as well as reduce the probability of entry errors and data loss. Accurate data collection and entry is necessary to the success of any research study. Similarly important is collection of item-level data rather than summary values. This allows researchers greater opportunity for discovery within a larger, more robust dataset (Nooner et al., 2012). Self Assessment enables researchers to collect and store all item-level assessment data in an efficient and timely way.

There are many facets to this tool that produce an easy-to-use interface and efficient data collection. Ease of use is one of the most important aspects considered while creating this tool - to reduce the time, energy, frustration of participants. The Self Assessment tool (SA) provides research staff an assessment queueing system, the ability to create user friendly instruments, the ability to review participant submitted assessments and easily export options. With this tool and others, COINS is striving to create an efficient, comprehensive and intuitive database to offer the research community.

METHODS

COINS OVERVIEW

COINS, created and developed at the Mind Research Network (MRN; The Mind Research Network for Neurodiagnostic Discovery, 2013), is a web-based data management system. COINS is unique in that it offers tools to collect, manage and share data of different modalities, including MRI, MEG, EEG and assessment data (Bockholt et al., 2010; Scott et al., 2011). There are similar neuroimaging suites (Marcus et al., 2007; Das et al., 2011), but they do not offer a module for participants to complete their own assessments.



To Add a New Study:

1) Enter the study details:
(All fields with * are required.)

* Study Name: Format: (XXXXXXXXXXXX) 12 characters max, matches name on scanner and archive

* IRB Title: Formal approved IRB title required.

* Principal Investigator: None □ Cross-Site PIs

Co-Investigator: None □ Cross-Site PIs

* Site: Mind Research Network

* Radiology Review Email List Format "email1@domain.edu, email2@domain.edu, " etc. Emails where radiology reviews go to.

* IRB Number Format "99-999" Default IRB Number to consent subjects under.

* Internal Study Number: Format "99-999" must be unique Correct number is crucial for receiving scans from the scanners

* Initial Approval Date: Format: "mm/dd/yyyy"

* Expiration Date: Format: "mm/dd/yyyy"

* Approved Participants:

* Allows URSI Sharing: No

* Expiration/Enrollment Warning Email List Format "email1@domain.edu, email2@domain.edu, " etc. Emails where study expiration warnings go

* Share Instrument Email List Format "email1@domain.edu, email2@domain.edu, " etc. Emails where instrument share requests go AND where notifications for shared instruments that are updated go

Sponsor: Optional: NIH for example

FIGURE 1 | Add a new study.

[1] SurveyMonkey Inc. Palo Alto, CA. Available online at: www.surveymonkey.com

[2] Qualtrics and all other Qualtrics product or service names are registered trademarks or trademarks of Qualtrics. Provo, UT. Available online at: http://www.qualtrics.com

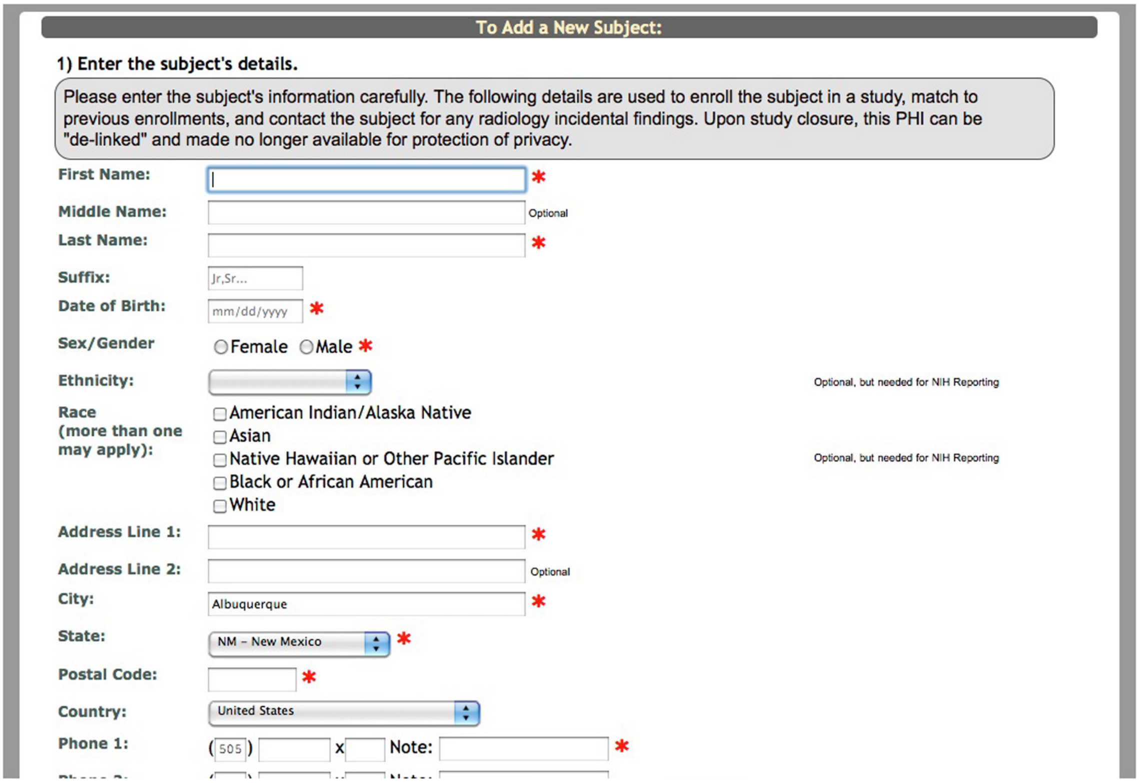
For the purpose of conveying where the Self Assessment tool fits into the study schema of COINS, we will briefly explain the process of adding a new study and enrolling participants in the system. Creating a study involves entering basic information about the study into a form (**Figure 1**). After this form has been submitted, the research staff must create study visits and subject types for the study. It is important to create study visits that reflect the study protocol. The study visits are used to associate assessments and scans with the time point on which the data was collected. Subject types are the different subject groups in the study protocol (e.g., Smoker, Non-Smoker).

After the study is set up, the research staff can begin enrolling participants into the study. Basic demographic information is entered for each participant during enrollment (**Figure 2**). At this time a subject type, chosen from the previously created subjects types, is assigned to the participant. Every time a participant is enrolled into a new study, they are assigned a study specific subject ID called an URSI (Unique Research Subject Identifier). In COINS all of the participant data (scans and assessments) are coded with the URSI and are linked together in data collection, data storage and at export (**Figure 3**).

PERMISSION LEVELS

The COINS database has been designed with security and restricted/controlled access features. External access to the system is restricted to either a VPN account with the Mind Research Network (MRN) or through a firewall rule for limited IP addresses within a collaborating institution. At the MRN, each user is given a dedicated, password protected COINS account that is only granted after all institutional human protections trainings have been completed. Account renewal is done annually and is dependent on human protections recertification. User access is based on study and role permissions. In order to gain access to a particular study, investigators need to have IRB approval to access that study within the database. The study PI will determine the level of study permissions the investigator is assigned based on their role (e.g., data entry, coordinator, co-investigator, etc.). Certain roles and applications allow an investigator to have access to a study, but not the participant identification details.

COINS is currently in compliance with HIPAA Privacy and Security Rule (Health Insurance Portability and Accountability Act of 1996, 2002) requirements. PHI is encrypted using the mcrypt libraries. The data exists on a virtual machine within the MRN firewall such that access to the machine is limited to only COINS system developers and IT personnel. Permission is granted by the site administrator on a granular level within each study. Raw data is restricted through user permissions both at the filesystem and the web application levels. Raw data, when viewed independent of the associated meta-data, is free of demographic PHI. PHI will be stored encrypted on a PostgresSQL database within the MRN network and protected by its firewall. Participant names and other identifying information will be maintained in this restricted database, available only to authorized members of the research team for the duration of the study. At the time of study closure, the link to participant names and other identifiable data will be unlinked and made inaccessible to the research team.



To Add a New Subject:

1) Enter the subject's details.

Please enter the subject's information carefully. The following details are used to enroll the subject in a study, match to previous enrollments, and contact the subject for any radiology incidental findings. Upon study closure, this PHI can be "de-linked" and made no longer available for protection of privacy.

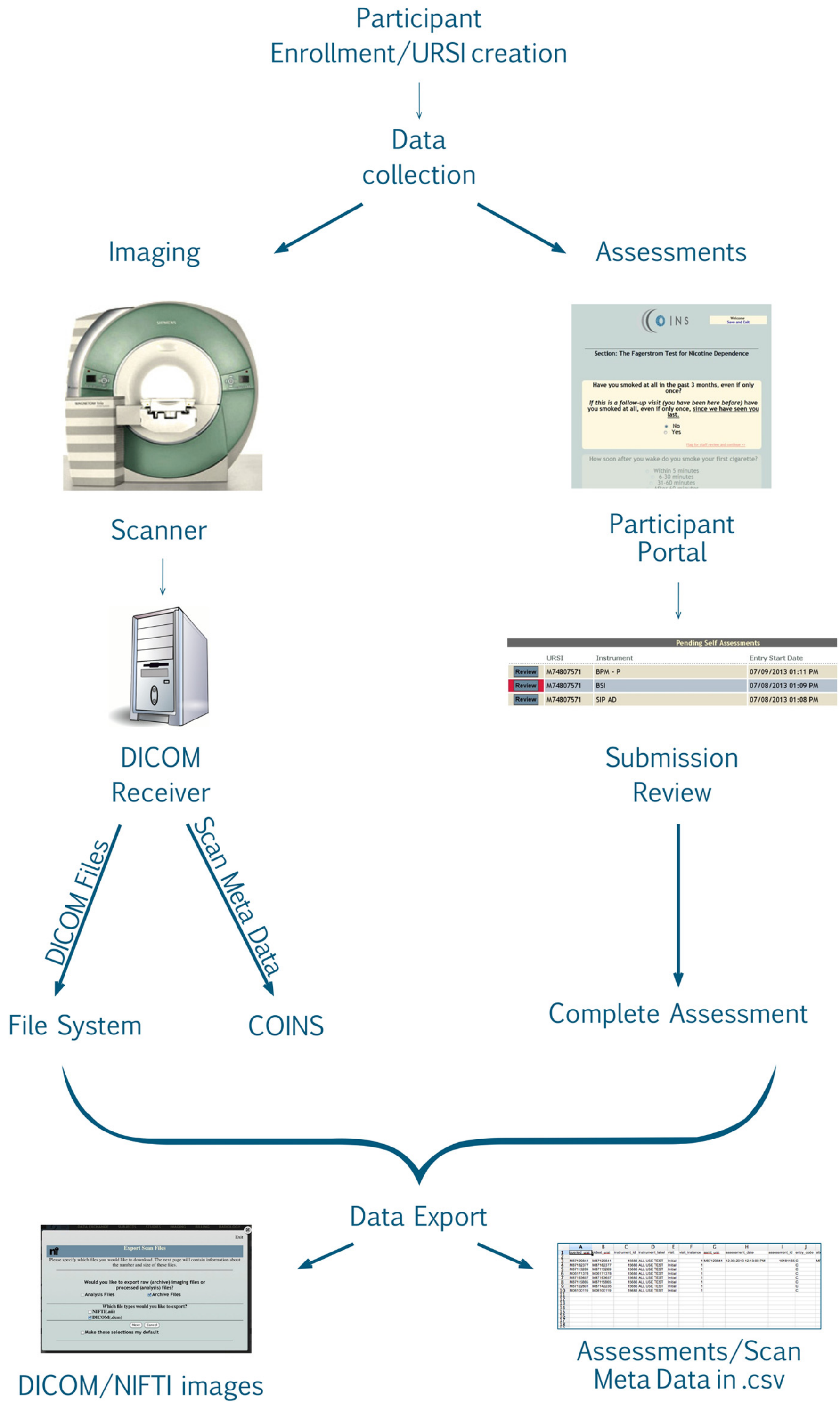
First Name: *
Middle Name: Optional
Last Name: *
Suffix: Jr,Sr...
Date of Birth: mm/dd/yyyy *
Sex/Gender ○Female ○Male *
Ethnicity: Optional, but needed for NIH Reporting
Race (more than one may apply): Optional, but needed for NIH Reporting
□American Indian/Alaska Native
□Asian
□Native Hawaiian or Other Pacific Islander
□Black or African American
□White
Address Line 1: *
Address Line 2: Optional
City: Albuquerque *
State: NM – New Mexico *
Postal Code: *
Country: United States
Phone 1: (505) x Note: *

FIGURE 2 | Add a new subject.

INSTRUMENTS

Clinical data gathered from interviews, questionnaires, and neuropsychological tests are entered into COINS through the Assessment Manager (ASMT) application. The Self Assessment tools are accessible through this application.

One of the first steps to assessment data collection in the COINS database is instrument creation. The term instrument here means the measure (or blank form) through which assessment data is entered (by the research staff or directly by the participants). Instruments can be created in several ways. There



COINS Welcome Save and Exit

Section: The Fagerstrom Test for Nicotine Dependence

Have you smoked at all in the past 3 months, even if only once?

If this is a follow-up visit (you have been here before) have you smoked at all, even if only once, since we have seen you last.

● No
○ Yes

Flag for staff review and continue >>

How soon after you wake do you smoke your first cigarette?

○ Within 5 minutes
○ 6-30 minutes
○ 31-60 minutes

Pending Self Assessments

	URSI	Instrument	Entry Start Date
Review	M74807571	BPM - P	07/09/2013 01:11 PM
Review	M74807571	BSI	07/08/2013 01:09 PM
Review	M74807571	SIP AD	07/08/2013 01:08 PM

Exit

Export Scan Files

Please specify which files you would like to download. The next page will contain information about the number and size of these files.

Would you like to export raw (archive) imaging files or processed (analysis) files?

□ Analysis Files ☑ Archive Files

Which file types would you like to export?

□ NIFTI(.nii)
☑ DICOM(.dcm)

Next Cancel

□ Make these selections my default

FIGURE 3 | Data collection flow chart.

is an instrument creation tool in which the general properties of the instrument are entered (instrument label, description, version, etc.), then sections are created manually as well as questions/responses. Another option to create an instrument is the instrument import tool. This process involves the research staff creating a template of the instrument in a.csv file that includes all of the fields that are required during manual instrument creation (instrument properties, sections, questions, etc.) which is then imported into the study by the COINS staff. The final option is to request an existing instrument be shared or copied to the investigator's study.

SELF ASSESSMENT QUESTION TYPES AND FEATURES

During instrument creation there are several features that can be employed to optimize the participant's experience. Instrument creation in COINS takes into account the need for participant friendly language. For this purpose there are Self Assessment specific instrument, section, and question labels available. These labels can be used in place of stigmatizing language that might influence a participant's responses.

This tool also has several different unique question types/features such as media question types. This question type can be used if the investigator would like to capture a participant's response to an image or video. The research staff can upload an image or video and create associated questions (multiple choice, visual analog scale, text response) (**Figure 4**). An extension of this question type is the continuous visual analog scale (VAS). Continuous VAS questions can be configured to record values at regular intervals while the participant is viewing a video. This can be used to allow participants to rate their emotional response to images/sounds in a video over time.

For the sake of efficiency and accuracy, COINS provides conditional looping, conditional skipping and auto-populated responses. Conditional looping and skipping (also sometimes referred to as "branching logic") allow the participant to move through a questionnaire without having to answer irrelevant questions (**Figure 5**). For example, a participant could skip out of answering cigarette smoking related questions if they do not smoke. Auto-populate questions can be used if more than one instrument asks the same question, for example, age. If the participant enters their age in the "auto-populate from" question, the "auto-populate to" question will display that response when the participant reaches that question. This reduces time of entry as well as frustration on the participant's part. The research staff can also choose to make the questions required, this option will not allow the participant to navigate away from a page until all questions are answered. This ensures that the assessment is fully completed. For questions that capture text responses, there are text enforcement options. These can be employed to be sure the correct type (date, phone number, number, time (HH:MM), etc.) of text response is entered (**Figure 6**).

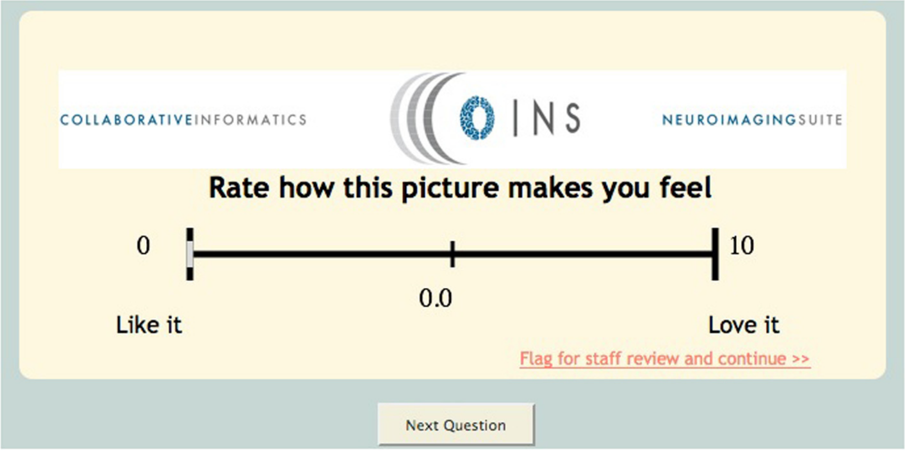
Often times neuroimaging research involves asking participants sensitive questions. The responses to these questions could lead to necessary intervention by the research staff (i.e., discussing suicidality). In order to alert the staff to such questions, there is a critical flagging feature that allows any response to be considered a critical flagged response. If such a response is selected by the participant, the research staff will see the assessment in red in their review queue (**Figure 7**), as well as the critical response question (**Figure 8**).

SELF ASSESSMENT PREVIEW

To ensure that the instrument presented to the participant functions (i.e., conditional skips, loops, specify options, etc.) as expected there is a tool called SA (Self Assessment) Preview. This tool allows research staff to view the instrument as a participant. Use of this tool is highly recommended for any instrument that will be viewed by participants. The research staff can view the instrument in SA Preview by the click of a button (available on each question), which launches a modal pop up. The instrument is displayed just as it would be to the participant.

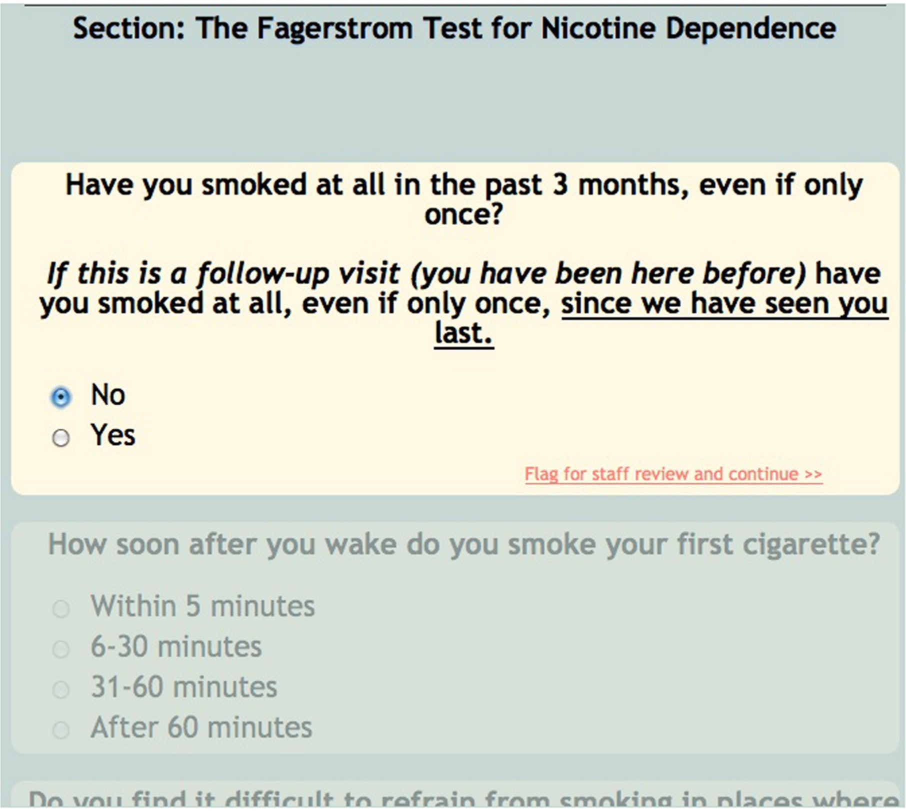
SELF ASSESSMENT QUEUES

In order for participants to fill out the assessments in the Participant Portal the research staff has to populate the "Participant Queue." Within ASMT there is a "Manage Subject



Rate how this picture makes you feel

FIGURE 4 | Visual analog scale.



Section: The Fagerstrom Test for Nicotine Dependence

Have you smoked at all in the past 3 months, even if only once?

If this is a follow-up visit (you have been here before) have you smoked at all, even if only once, since we have seen you last.

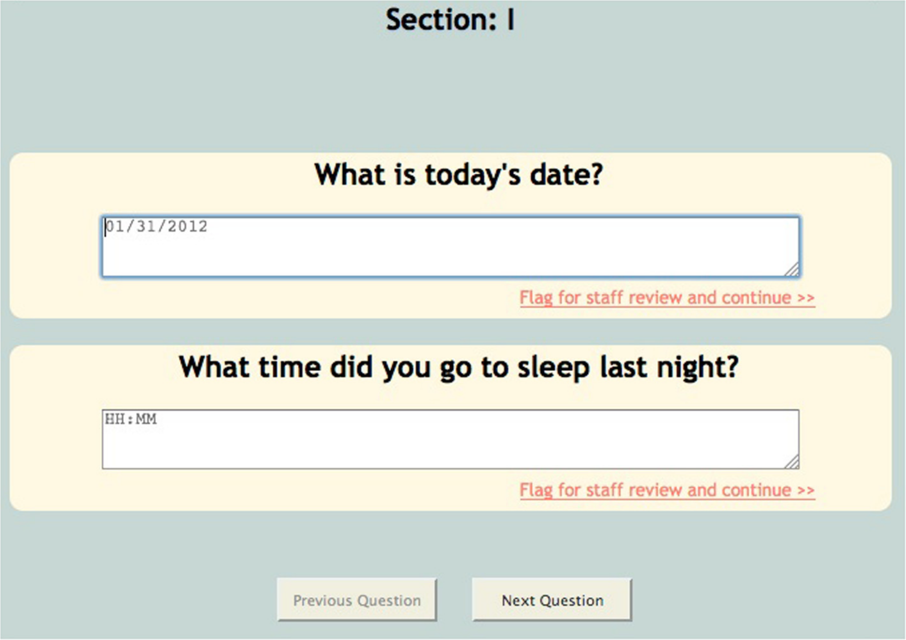
● No
○ Yes

How soon after you wake do you smoke your first cigarette?

○ Within 5 minutes
○ 6-30 minutes
○ 31-60 minutes
○ After 60 minutes

Do you find it difficult to refrain from smoking in places where

FIGURE 5 | Conditional skipping.



Section: I

What is today's date?

01/31/2012

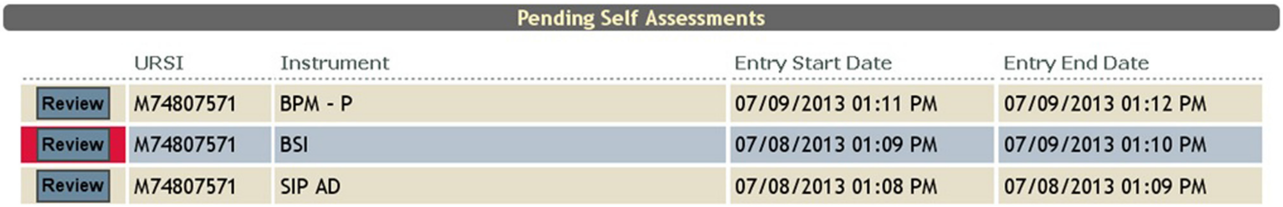
Flag for staff review and continue >>

What time did you go to sleep last night?

HH:MM

Flag for staff review and continue >>
Previous Question
Next Question

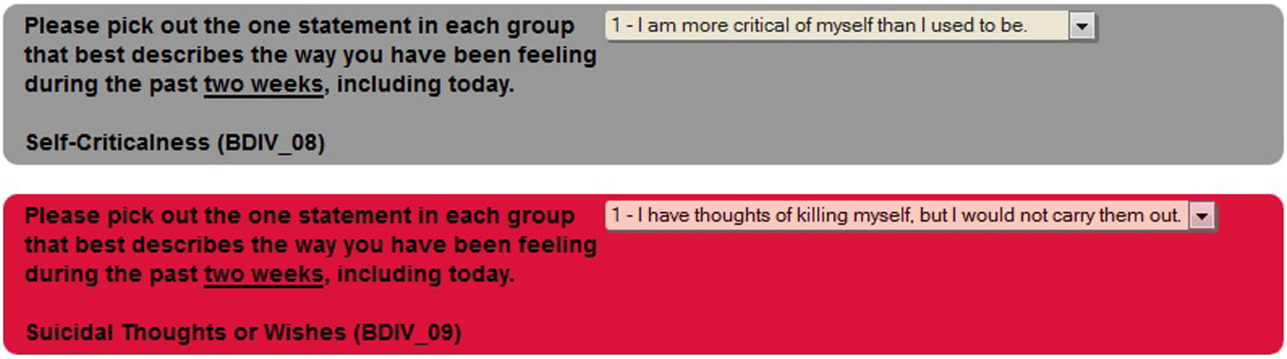
FIGURE 6 | Text enforcement.



Pending Self Assessments

	URSI	Instrument	Entry Start Date	Entry End Date
Review	M74807571	BPM - P	07/09/2013 01:11 PM	07/09/2013 01:12 PM
🟥 Review	M74807571	BSI	07/08/2013 01:09 PM	07/09/2013 01:10 PM
Review	M74807571	SIP AD	07/08/2013 01:08 PM	07/08/2013 01:09 PM

FIGURE 7 | Critical flagging in review queue.



Please pick out the one statement in each group that best describes the way you have been feeling during the past two weeks, including today.

1 - I am more critical of myself than I used to be.

Self-Criticalness (BDIV_08)

Please pick out the one statement in each group that best describes the way you have been feeling during the past two weeks, including today.

1 - I have thoughts of killing myself, but I would not carry them out.

Suicidal Thoughts or Wishes (BDIV_09)

FIGURE 8 | Critical flagged question.

Queues" tool. The user selects the participant's URSI, the study visit (e.g., Baseline, Visit 1, Visit 2) for which they want data entered, the queue type and then creates a login for the participant (it is recommended that these logins not contain any participant identifiers). Queue types determine how the assessment is handled in the Participant Portal. A one-time queue is used for assessments that are only to be collected once per visit. An on-going queue is used if the assessment data is collected throughout the study (e.g., calendar data). Once the data collection is over only the research staff can complete an assessment in the on-going queue. A recurring queue is used if an assessment needs to be collected more than once per visit. Each time the participant opens an assessment in this queue a new assessment is begun and an new instance is created in the database. As with the on-going queue only the research staff can complete an assessment in this queue type.

Creating a queue is a drag and drop system. The interface displays a box for the "Participant Queue," "Study Templates," and "Study Instruments" (**Figure 9**). The "Study Instruments" box includes all of the instruments that have been created for the study. To populate the queue the research staff has to click each desired instrument and drag it from the "Study Instruments" box to the "Participant Queue" box and release. When all instruments have been queued, they save the list and can provide the participant with the website (coins.mrn.org/p2) and login.

TEMPLATES

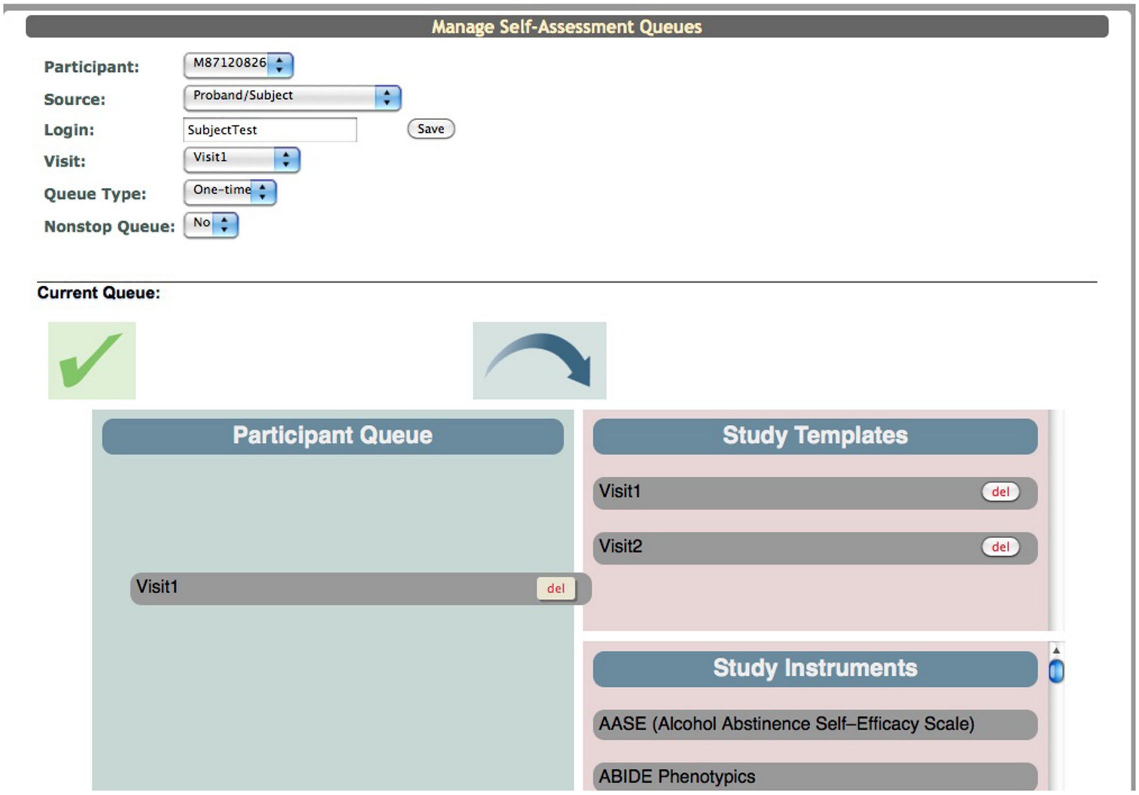
A template schema was created for ease of use. The user can drag and drop all of the instruments into the "Participant Queue," click a button and a pop up appears that asks for a template name. The template then appears in the "Study Templates" box. When the next participant is ready to be queued for assessments, the research staff can drag the template previously created over to the "Participant Queue" and the instruments will appear in the same order in which they were saved. This reduces the amount time that it takes for the user to set up the queue as well as accounts for any potential error (forgetting an instrument, adding two of the same instrument, etc.).

PARTICIPANT PORTAL

A participant can begin filling out the queued assessments as soon as they login into the Participant Portal. The Participant Portal can be accessed anywhere with an internet connection. The portal has been designed to have an easy to use interface for all ranges of participant types, from those that are computer savvy to those that have had little exposure to computers.

As the participant is completing the assessments, they are made aware of their progress. At the bottom of the screen there is a note to the user indicating how many assessments they have to complete. Also at the end of each assessment there is a brief message that they have completed the assessment and indicates how many assessments are left in their queue. If the participant needs a break, they can click "Save and Exit" and when they log back in they will be brought the last unanswered question.

As the participants complete the assessments the research staff receive emails indicating that an assessment is complete and waiting to be reviewed in the review queue. These emails also contain a link to the review queue so that the researcher can easily access it. In order to receive the notification emails, the research



Manage Self-Assessment Queues

Participant: M87120826

Source: Proband/Subject

Login: SubjectTest

Visit: Visit1

Queue Type: One-time

Nonstop Queue: No

Current Queue:

Participant Queue

Visit1

Study Templates

Visit1

Visit2

Study Instruments

AASE (Alcohol Abstinence Self–Efficacy Scale)

ABIDE Phenotypics

FIGURE 9 | Participant Queue.

staff enter the desired email addresses in a list during study set up. This list can be edited throughout the duration of the study so that only those who need to, receive the assessment emails.

CUSTOMIZED CSS IN SELF ASSESSMENT

At the site level, users can customize the Participant Portal with a CSS upload tool. With a basic understanding of CSS, users create a CSS file to change the layout, background, color, and fonts of any generic element or specify a class or id to change more specific elements. They can also upload a logo or graphic from their institution to be displayed at the top of every page in Self Assessment. These tools provide the participant with a feeling of continuity as the Participant Portal will have the same look and feel of the other websites that they are using during their study participation (**Figure 10**).

SELF ASSESSMENT REVIEW QUEUE

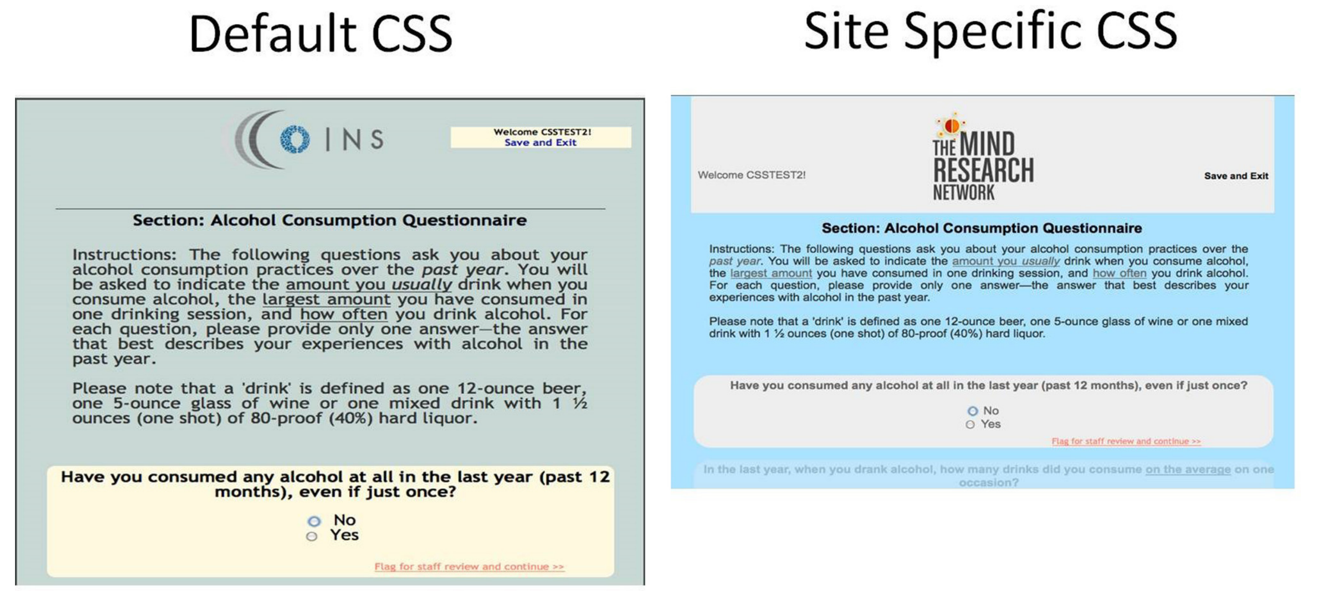
The review queue contains all of the self assessments that have been completed by participants. The research staff member reviewing the assessments has the option to complete the assessment, deny the assessment or save the assessment to review later. If there are no issues with the assessment the study staff can click "Complete" to send the assessment to the database as a finished, complete record (no further entry is needed). If the assessment is incomplete or there is a response that needs clarification, the assessment can be denied. When an assessment is denied it is sent back to the participant's queue for completion/updating. If the user cannot complete the review, they can save it and escape the assessment in order to keep it in their review queue to be reviewed and completed/denied at another time.

The Self Assessment time log is a tool to determine how long participants spend on individual questions or pages while completing an assessment in the Participant Portal. There is a list of all of the self assessments for the study. Included on that list is a column labeled, "time spent," which displays the time, in minutes, that it took the participant to complete that specific assessment. The user can also view a further breakdown of the time log that displays timing information on every event completed in the assessment (e.g., assessment resumed, question answered, next page button clicked, assessment complete, etc.) (**Figure 11**).

EXPORTING DATA

Data collected via SA is easily retrieved and exported from Query Builder and/or a study portal. Query Builder is the most versatile data export tool currently offered in the COINS tool suite. This tool supports secure, *ad hoc* querying of single and cross-site studies for assessments, scans and demographics. It also offers the ability to search assessment data and scan data in the same query.

A study portal is a centralized collaboration tool for monitoring enrollment progress, quality assurance, document exchange, etc. Progress reports within the portals provide a complete workflow overview of a study to identify missing data at one glance. Internal and external collaborators can



Welcome CSSTEST2!
Save and Exit

Section: Alcohol Consumption Questionnaire

Instructions: The following questions ask you about your alcohol consumption practices over the *past year*. You will be asked to indicate the amount you *usually* drink when you consume alcohol, the largest amount you have consumed in one drinking session, and how often you drink alcohol. For each question, please provide only one answer—the answer that best describes your experiences with alcohol in the past year.

Please note that a 'drink' is defined as one 12-ounce beer, one 5-ounce glass of wine or one mixed drink with 1 ½ ounces (one shot) of 80-proof (40%) hard liquor.

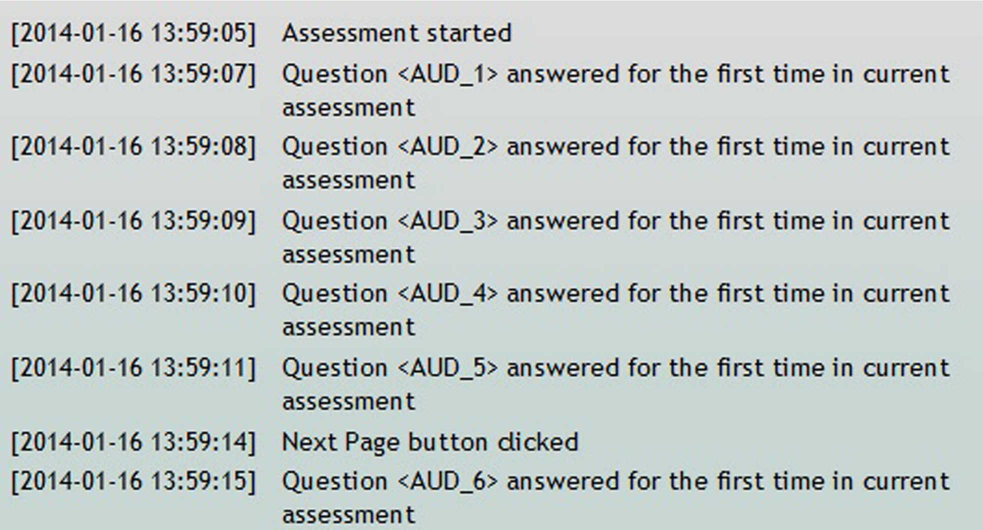
Have you consumed any alcohol at all in the last year (past 12 months), even if just once?

○ No
○ Yes

Flag for staff review and continue >>

In the last year, when you drank alcohol, how many drinks did you consume on the average on one occasion?

FIGURE 10 | Participant Portal.



```
[2014-01-16 13:59:05]  Assessment started
[2014-01-16 13:59:07]  Question <AUD_1> answered for the first time in current
                       assessment
[2014-01-16 13:59:08]  Question <AUD_2> answered for the first time in current
                       assessment
[2014-01-16 13:59:09]  Question <AUD_3> answered for the first time in current
                       assessment
[2014-01-16 13:59:10]  Question <AUD_4> answered for the first time in current
                       assessment
[2014-01-16 13:59:11]  Question <AUD_5> answered for the first time in current
                       assessment
[2014-01-16 13:59:14]  Next Page button clicked
[2014-01-16 13:59:15]  Question <AUD_6> answered for the first time in current
                       assessment
```

FIGURE 11 | Self assessment time log.

use the portals to access all assessment data associated with their studies. Assessment data can be exported by subject type/instrument/study visit. There are also reports that display graphical views of question scores, demographic statistics and outliers in the data.

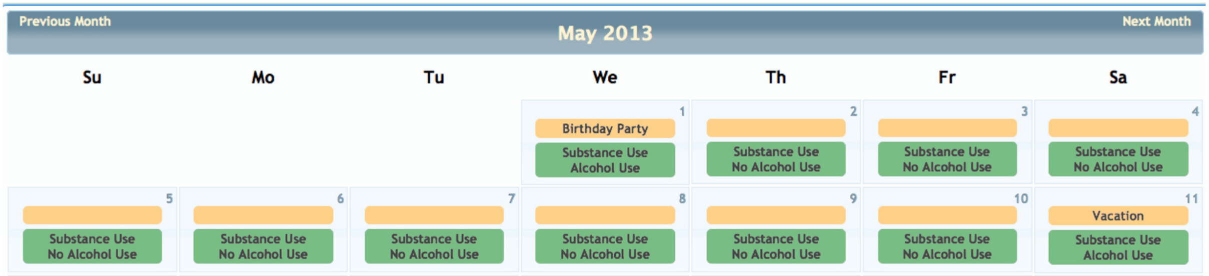
RESULTS

Since the release of the Self Assessment tool in 2011, there have been 35,448 assessments collected across 6 sites (**Table 1**). Several studies/programs have been instrumental in the continual development of this tool. The enhanced Nathan Kline Institute - Rockland Sample (NKI-RS) is an ongoing project aimed at collecting 1,000 or more participants to provide a lifespan sample (ages 6–85 years old) of phenotypic, neuroimaging and genetics data (Nooner et al., 2012). The initial development of Self Assessment was guided by the expected types of assessments collected by the NKI-RS project. Currently, NKI-RS almost exclusively collects assessment data via Self Assessment, sometimes collecting over 15 assessments at one visit.

Although COINS allows cross modality data collection, not every group using COINS collects imaging data. The New Mexico Works Intensive Case Management, Recovery and Employment (ICARE) program is a pilot program designed to address substance use barriers to employment in Temporary Assistance for Needy Families (TANF) recipients (NM Human Services Department, 2012). The substance abuse data is collected via an SA calendar tool that has been tailored to the Timeline Followback assessment (Sobell and Sobell, 1992). This particular calendar tool is designed to collect life events as well as substance use information for a complete emulation of a paper and pencil Timeline Followback assessment. Data entered by the day into the virtual calendar (**Figure 12**) can be duplicated, edited and deleted. Multiple days with same information can be entered all at once via simple key commands. This tool is capable of continuous entry when queued in an On-going queue type. The Followback Calendar includes an administrator tool that allows staff to edit previously entered data in the event of an entry error or incorrect reporting. The information entered into the calendar through Self Assessment can then be viewed and exported in very detailed and easy-to-use reports. Substance use information from the calendar can be graphically viewed in several different charts types (**Figure 13**) via the "Calendar Report Tool." Each unique substance reported on the calendar can be shown or hidden with toggle icons and can be viewed as a simple, clean bar graph or as a cumulation graph, showing length of use and abstinence periods. All life events and substances used are also plainly listed by

Table 1 | Assessments per site.

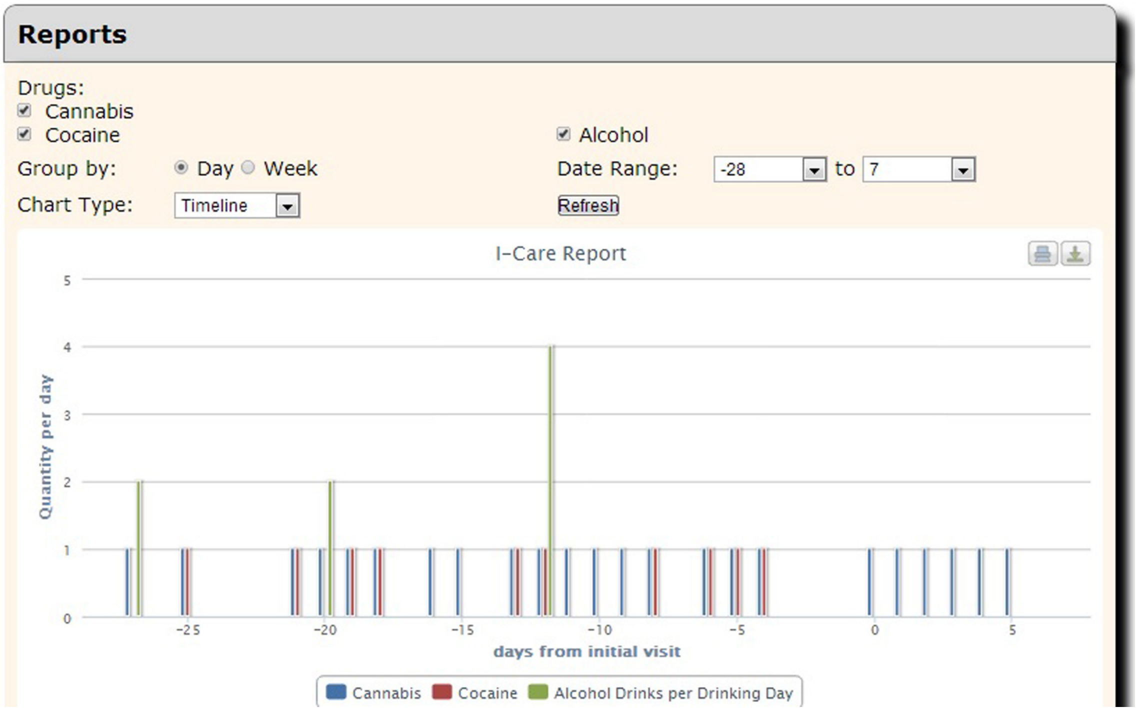
Sites	Number of assessments
The Mind Research Network	20,613
Nathan Kline Institute	12,136
NM works—ICARE	2166
University of North Carolina—Wilmington	286
University of Colorado Boulder	238



Previous Month	May 2013	Next Month

Su	Mo	Tu	We	Th	Fr	Sa
			1 Birthday Party Substance Use Alcohol Use	2 Substance Use No Alcohol Use	3 Substance Use No Alcohol Use	4 Substance Use No Alcohol Use
5 Substance Use No Alcohol Use	6 Substance Use No Alcohol Use	7 Substance Use No Alcohol Use	8 Substance Use No Alcohol Use	9 Substance Use No Alcohol Use	10 Substance Use No Alcohol Use	11 Vacation Substance Use Alcohol Use

FIGURE 12 | ICARE calendar.



Reports

Drugs:
☑ Cannabis
☑ Cocaine
☑ Alcohol

Group by: ● Day ○ Week

Date Range: -28 to 7

Chart Type: Timeline

Refresh

FIGURE 13 | Calendar report tool.

date for easy review. In addition to the ability to effortlessly visualize the substance use data, it can also be exported for analysis, allowing for the day range, days per interval customization. This project serves participants that have little or no knowledge of computers and thus far there have not been any barriers during their use of the tool. There have been 2166 self assessments collected for this project thus far.

DISCUSSION

COINS is under constant development in order to satisfy the need for a database that provides tools for all aspects of a research study. Although we offer a robust tool suite there are several areas in which we can improve and provide more features.

AUTO-QUEUES

We plan to continue to reduce ways in which the research staff have to manually enter information. We are currently developing an auto assessment-queuing process. This tool will enable research staff to set up conditions for automated assessment queues (currently done manually by research staff) based on subject types and/or responses to specific questions. This will reduce errors (queuing incorrect assessments, queuing for incorrect visits, etc.) and the amount of time spent by research staff.

OFFLINE DATA STORAGE

Data collection is often conducted in the field, where wireless internet can be unpredictable or non-existent. COINS currently offers a Windows-XP-Tablet-based direct entry application that uses a web service to sync assessment data to the database when a data connection is available (Turner et al., 2011). This application is primarily used by research studies that need to collect data in an environment where no data connection is available (e.g., correctional institutions or rural populations). This tool has proven extremely useful for this purpose and requires little maintenance after a study has been set up. Unfortunately, the application was designed for use exclusively on touch-based Windows XP devices. These devices will no longer be supported by Microsoft in the spring of 2014, and newer tablet technology from Apple, Microsoft and Google warrants a new offline-capable system.

To this end, we plan to leverage HTML5 web standards such as the Local Storage API (Hickson, 2013), and Cache Manifest (HTML5, 2012). This will allow any device with a browser to cache instruments for use in an offline environment, and then store data entered into those instruments on the device until a data connection can be established. Once a data connection is established, data will be synced to the COINS servers, where it can be inspected, approved and imported.

CONCLUSION

There are several options available to researchers for assessment data collection, but very few that offer a full neuroimaging tool suite as well as participant entered assessments. The COINS Self Assessment tool is optimal for participant data collection due to its ease of use (for participants and research staff), integration capability with other neuroimaging data, security features for protecting sensitive/identifying participant information. The COINS team will continue to improve the usability of current tools as well as aim to provide new features and tools that will allow COINS stand out as a superior alternative to collecting study data with several different databases/systems.

REFERENCES

Bockholt, H. J., Scully, M., Courtney, W., Rachakonda, S., Scott, A., Caprihan, A., et al. (2010). Mining the mind research network: a novel framework for exploring large scale, heterogeneous translational neuroscience research data sources. *Front. Neuroinform.* 3:36. doi: 10.3389/neuro.11.036.2009

Buhrmester, M., Kwang, T., and Gosling, S. D. (2011). Amazon's mechanical turk: a new source of inexpensive, yet high-quality, data? *Perspect. Psychol. Sci.* 6, 3–5. doi: 10.1177/1745691610393980

Das, S., Zijdenbos, A. P., Harlap, J., Vins, D., and Evans, A. C. (2011). LORIS: a web-based data management system for multi-center studies. *Front. Neuroinform.* 5:37. doi: 10.3389/fninf.2011.00037

Health Insurance Portability and Accountability Act of 1996. (2002). *45 CFR Part 160 and Subparts A and E of Part 164.*

Hickson, I. (2013). *World Wide Web Consortium Recommendation: Web Storage.* Available online at: http://www.w3.org/TR/webstorage/ (Accessed January 31, 2014).

HTML5. (2012). *A Vocabulary and Associated APIs for HTML and XHTML, Editor's Draft 22 August 2012, 5.7 Offline Web Applications, World Wide Web Consortium.* Available online at: http://dev.w3.org/html5/spec-preview/offline.html (Accessed January 31, 2014).

Marcus, D. S., Olsen, T. R., Ramaratnam, M., and Buckner, R. L. (2007). The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34. doi: 10.1385/NI:5:1:11

Mind Research Network. (2013). *A Nonprofit 501(c)3 Organization, Partnered With Lovelace Respiratory Research Institute.* Available online at: www.mrn.org

NM Human Services Department. (2012). *NM Works I-CARE Program.* Available online at: http://www.hsd.state.nm.us/LookingForAssistance/i-care.aspx (Accessed January 31, 2014).

Nooner, K. B., Colcombe, S. J., Tobe, R. H., Mennes, M., Benedict, M. M., Moreno, A. L., et al. (2012). The NKI-Rockland sample: a model for accelerating the pace of discovery science in psychiatry. *Front. Neurosci.* 6:152. doi: 10.3389/fnins.2012.00152

Scott, A., Courtney, W., Wood, D., de la Garza, R., Lane, S., King, M., et al. (2011). COINS: an innovative informatics and neuroimaging tool suite built for large heterogeneous datasets. *Front. Neuroinform.* 5:33. doi: 10.3389/fninf.2011.00033

Sobell, L. C., and Sobell, M. B. (1992). "Timeline follow-back: a technique for assessing self-reported alcohol consumption," in Measuring *Alcohol Consumption: Psychosocial and Biological Methods*, eds R. A. Litten and J. P. Allen (Totowa, NJ: Humana Press), 41–72.

Turner, J. A., Lane, S. R., Bockholt, H. J., and Calhoun, V. D. (2011). The clinical assessment and remote administration tablet. *Front. Neuroinform.* 5:31. doi: 10.3389/fninf.2011.00031

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 03 February 2014; accepted: 15 May 2014; published online: 05 June 2014.
Citation: King MD, Wood D, Miller B, Kelly R, Landis D, Courtney W, Wang R, Turner JA and Calhoun VD (2014) Automated collection of imaging and phenotypic data to centralized and distributed data repositories. Front. Neuroinform. ***8****:60. doi: 10.3389/fninf.2014.00060*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 28 May 2014
doi: 10.3389/fninf.2014.00058

Extending the NIF DISCO framework to automate complex workflow: coordinating the harvest and integration of data from diverse neuroscience information resources

***Luis N. Marenco*[1,2,3], *Rixin Wang*[1], *Anita E. Bandrowski*[4], *Jeffrey S. Grethe*[4], *Gordon M. Shepherd*[3]* and *Perry L. Miller*[1,2,5,6]**

[1] Center for Medical Informatics, Yale University School of Medicine, New Haven, CT, USA
[2] VA Connecticut Healthcare System, US Department of Veterans Affairs, West Haven, CT, USA
[3] Department of Neurobiology, Yale University School of Medicine, New Haven, CT, USA
[4] Department of Neurosciences, Center for Research in Biological Systems, University of California at San Diego, La Jolla, CA, USA
[5] Department of Anesthesiology, Yale University School of Medicine, New Haven, CT, USA
[6] Department of Molecular, Cellular and Developmental Biology, Yale University, New Haven, CT, USA

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Graham J. Galloway, The University of Queensland, Australia
Yaroslav O. Halchenko, Dartmouth College, USA

Correspondence:
Gordon M. Shepherd, Department of Neurobiology, Yale University School of Medicine, P.O. Box 208001, New Haven, CT 06520-8001, USA
e-mail: gordon.shepherd@yale.edu

This paper describes how DISCO, the data aggregator that supports the Neuroscience Information Framework (NIF), has been extended to play a central role in automating the complex workflow required to support and coordinate the NIF's data integration capabilities. The NIF is an NIH Neuroscience Blueprint initiative designed to help researchers access the wealth of data related to the neurosciences available via the Internet. A central component is the NIF Federation, a searchable database that currently contains data from 231 data and information resources regularly harvested, updated, and warehoused in the DISCO system. In the past several years, DISCO has greatly extended its functionality and has evolved to play a central role in automating the complex, ongoing process of harvesting, validating, integrating, and displaying neuroscience data from a growing set of participating resources. This paper provides an overview of DISCO's current capabilities and discusses a number of the challenges and future directions related to the process of coordinating the integration of neuroscience data within the NIF Federation.

Keywords: data integration, database federation, database interoperation, neuroinformatics, biomedical informatics

INTRODUCTION

Experimental and computational data in neuroscience increasingly overwhelms our ability to integrate it to give insight into the molecular and cellular basis of normal and diseased neuronal function. The problem is extreme in neuroscience because the data comes from a wide variety of disciplines. Tools are therefore urgently needed for automating the discovery, extraction, and organization of this data. This paper describes the current status of the DISCO framework that has been extended to play a central role in automating the complex workflow required to support and coordinate the data integration capabilities of the Neuroscience Information Framework (NIF). The NIF[1] is an NIH Neuroscience Blueprint initiative designed to help researchers access the wealth of data related to the neurosciences available via the Internet (Gardner et al., 2008; Gupta et al., 2008; Bandrowski et al., 2012; Cachat et al., 2012). A central component is the NIF Federation, a searchable database that currently (as of January, 2014) contains data that is downloaded on an ongoing basis from over 231 data and information resources (for an updated list see, http://disco.neuinfo.org).

A user querying the NIF Federation typically receives results from a range of resources containing data relevant to the query. For most resources, further information about a particular data item can be obtained by linking directly to data stored within the resource itself.

The NIF Federation is growing as new resources are added, and as new data are downloaded from participating resources. A major challenge involves the need to keep the data contained within the NIF Federation up-to-date, since most of its information resources are accumulating new data on a regular basis that need to be downloaded to the NIF. In addition, data previously downloaded from a resource may need to be changed to reflect changes made to the data within the resource. Furthermore, the internal structure of a resource may periodically change, requiring that the logic that "harvests" data from that resource be modified.

DISCO (DISCOvery) was initially developed as a set of tools to assist in focused aspects of the process described above (Marenco et al., 2010). During the past several years the role of DISCO has expanded dramatically to play a central role in automating the complex data-pipeline workflow required. Examples of DISCO's capabilities include the following.

- creating a new data resource in the NIF Federation describing what data to extract and how to extract that data,

[1] www.neuinfo.org

- setting up a schedule for downloading new data from a resource,
- downloading the current data from a resource and comparing it to the previous version of that data if one exists,
- creating a new version of the data for a resource and putting it in a temporary ("beta") file to allow it to be inspected and approved before it is officially loaded into the operational version of the NIF Federation,
- allowing the NIF staff to create views of the NIF Federation data with the help of a concept mapper, including integrated views that combine data from multiple resources,
- alerting the NIF staff if problems are detected in any of these activities, and helping coordinate the resolution of each problem,
- maintaining a record of all these activities as they occur.

This paper provides an overview of DISCO's current capabilities and discusses some of the issues and challenges involved in coordinating the integration of neuroscience data within the NIF Federation.

BACKGROUND

DISCO can be described as an extensible data aggregator designed to facilitate automated information integration from disparate data sources over time. To help accomplish this goal DISCO includes the following features: persistence of provenance storage representation, historical data tracking, semantic data mappings, and near real-time federated data synchronization.

The most commonly known aggregators are Web crawlers. These scan the content of Web pages on a regular basis and index the terms extracted from free text retrieved. Any data that is stored inside Web-accessible databases, however, is not scanned and is therefore "invisible" to Web crawlers. DISCO differs in that it uses resource-specific tailored logic to guide focused data extraction from a variety of Web-based data-presentation formats, including Internet-accessible databases.

Two general approaches that have been widely described for integrating data from multiple distributed databases are (1) a data warehouse approach and (2) a data federation approach. In a data warehouse, data from participating resources are downloaded to a central database where they can be queried locally in an integrated fashion. Examples of data warehouses in the life sciences and clinical medicine includes DWARF (Fischer et al., 2006) and i2b2 clinical data warehouse (Majeed and Röhrig, 2012). By contrast, in a data federation, the data is not downloaded to a central database but remains stored within each participating resource. The federation (1) allows the user to submit a query, (2) breaks that query down into a set of individual subqueries that are submitted to each appropriate resource, and (3) integrates the results returned. Examples of biomedical data federated systems include InterPro BioMart (Jones et al., 2011) and caGrid (Saltz et al., 2006).

The NIF Federation implements a hybrid approach. A central component of the NIF Federation is a searchable data warehouse that contains selected data elements from participating resources. A major advantage of this approach is that queries can be executed much faster since all the data is stored in one place. There are no network communication latencies and no issues of participating resources being temporarily unavailable. In addition, complex database joins can be particularly difficult to implement within a pure data federation when the results of intermediate database operations need to be transmitted over the network. The NIF Federation is a hybrid because in addition to the data stored in its data warehouse, a large amount of additional data stored in individual resources can be linked to from data elements stored in the warehouse. This allows the user to "drill down" to very detailed data (for example to raw data such as complex experimental results) once the user determines based on a search of the warehouse that this additional data is of potential interest.

Many data warehouses store the data they retrieve from participating databases in a normalized form, so that the combined data can be queried as one single source. The NIF Federation does not do this. Normalizing data from highly dynamic heterogeneous federation of resources would be prohibitively time-consuming, if not impossible. As a result, the NIF stores the data retrieved from each participating resource in separate resource-specific tables.

DISCO stores the data that it extracts from all resources in relational form. Each resource has its data stored in its own set of DISCO tables. This approach to data extraction and storage forms the foundation for DISCO that facilitates data integration, searching over multiple resources, and tracking over time. Integrated querying of content within the NIF Federation is supported by first examining the text content and annotation of the imported data and mapping these to NIFSTD, the NIF ontology (Bug et al., 2008; Imam et al., 2012). Queries can then be expressed using NIFSTD. In addition, the NIF Federation has established some integrated views of data from multiple resources that allow those data to be more tightly linked for query purposes.

OVERVIEW OF THE DISCO FUNCTIONALITY

This section first provides an overview of DISCO's capabilities, after which we describe DISCO's operation in more detail using a number of Web screens that illustrate concretely various aspects of DISCO's functionality. More detail about the function of DISCO as a whole can be found in the online DISCO User's and Technical Manual[2].

The current DISCO implementation represents a major advance over an early version of DISCO previously described in (Marenco et al., 2010). The previous version provided an initial set of tools that helped participating resource staff define the data they wished to include in the NIF Federation, and the upload of that data to the Federation. In its current implementation, DISCO provides a sophisticated infrastructure that coordinates and orchestrates three separate, but interrelated federated data processing pipelines for the NIF. The overall process coordinated by these pipelines is illustrated in **Figure 1**.

- The first pipeline involves DISCO resource registration. This process is initiated by creating a new resource sitemap at NeuroLex (neurolex.org). At the end of this initial step, this information is pushed to DISCO. Once registered in DISCO, NIF data curators then work with resource staff

[2] http://disco.neuinfo.org/docs/manual/

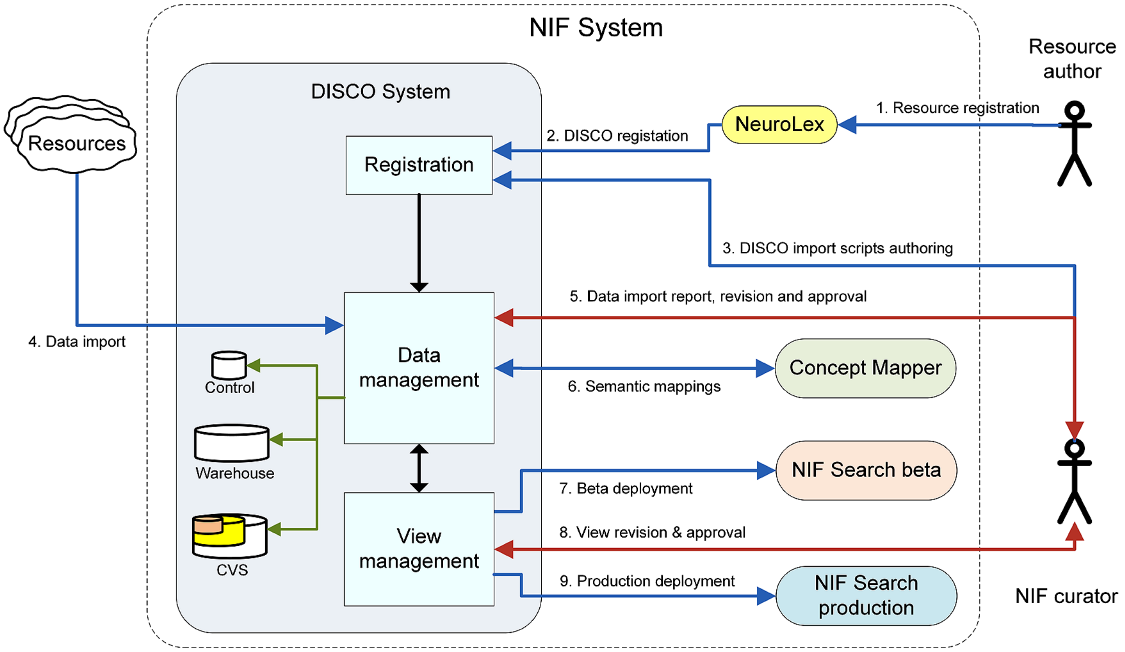
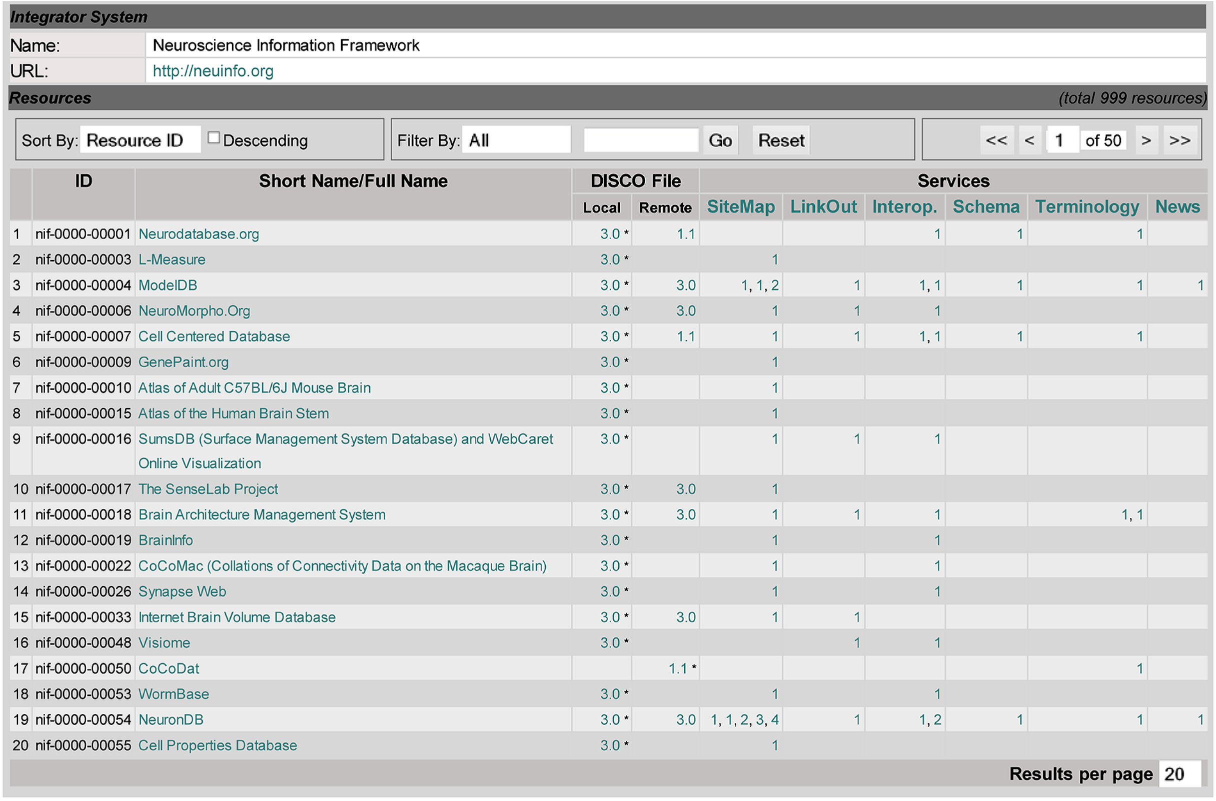


FIGURE 1 | A schematic overview of DISCO's functionality.



Integrator System

Name:	Neuroscience Information Framework
URL:	http://neuinfo.org

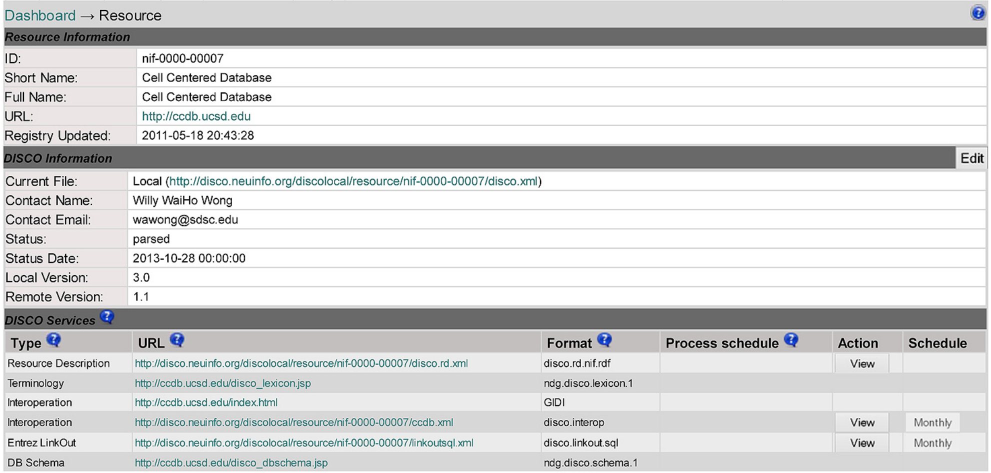
Resources *(total 999 resources)*

Sort By: Resource ID □Descending Filter By: All Go Reset << < 1 of 50 > >>

	ID	Short Name/Full Name	DISCO File		Services					
			Local	Remote	SiteMap	LinkOut	Interop.	Schema	Terminology	News
1	nif-0000-00001	Neurodatabase.org	3.0 *	1.1			1	1	1	
2	nif-0000-00003	L-Measure	3.0 *		1					
3	nif-0000-00004	ModelDB	3.0 *	3.0	1, 1, 2	1	1, 1	1	1	1
4	nif-0000-00006	NeuroMorpho.Org	3.0 *	3.0	1	1	1			
5	nif-0000-00007	Cell Centered Database	3.0 *	1.1	1	1	1, 1	1	1	
6	nif-0000-00009	GenePaint.org	3.0 *		1					
7	nif-0000-00010	Atlas of Adult C57BL/6J Mouse Brain	3.0 *		1					
8	nif-0000-00015	Atlas of the Human Brain Stem	3.0 *		1					
9	nif-0000-00016	SumsDB (Surface Management System Database) and WebCaret Online Visualization	3.0 *		1	1	1			
10	nif-0000-00017	The SenseLab Project	3.0 *	3.0	1					
11	nif-0000-00018	Brain Architecture Management System	3.0 *	3.0	1	1	1		1, 1	
12	nif-0000-00019	BrainInfo	3.0 *		1		1			
13	nif-0000-00022	CoCoMac (Collations of Connectivity Data on the Macaque Brain)	3.0 *		1		1			
14	nif-0000-00026	Synapse Web	3.0 *		1		1			
15	nif-0000-00033	Internet Brain Volume Database	3.0 *	3.0	1	1				
16	nif-0000-00048	Visiome	3.0 *			1	1			
17	nif-0000-00050	CoCoDat		1.1 *					1	
18	nif-0000-00053	WormBase	3.0 *		1		1			
19	nif-0000-00054	NeuronDB	3.0 *	3.0	1, 1, 2, 3, 4	1	1, 2	1	1	1
20	nif-0000-00055	Cell Properties Database	3.0 *		1					

Results per page 20

FIGURE 2 | The main DISCO Dashboard, as described in the text.



Dashboard → Resource

Resource Information

ID:	nif-0000-00007
Short Name:	Cell Centered Database
Full Name:	Cell Centered Database
URL:	http://ccdb.ucsd.edu
Registry Updated:	2011-05-18 20:43:28

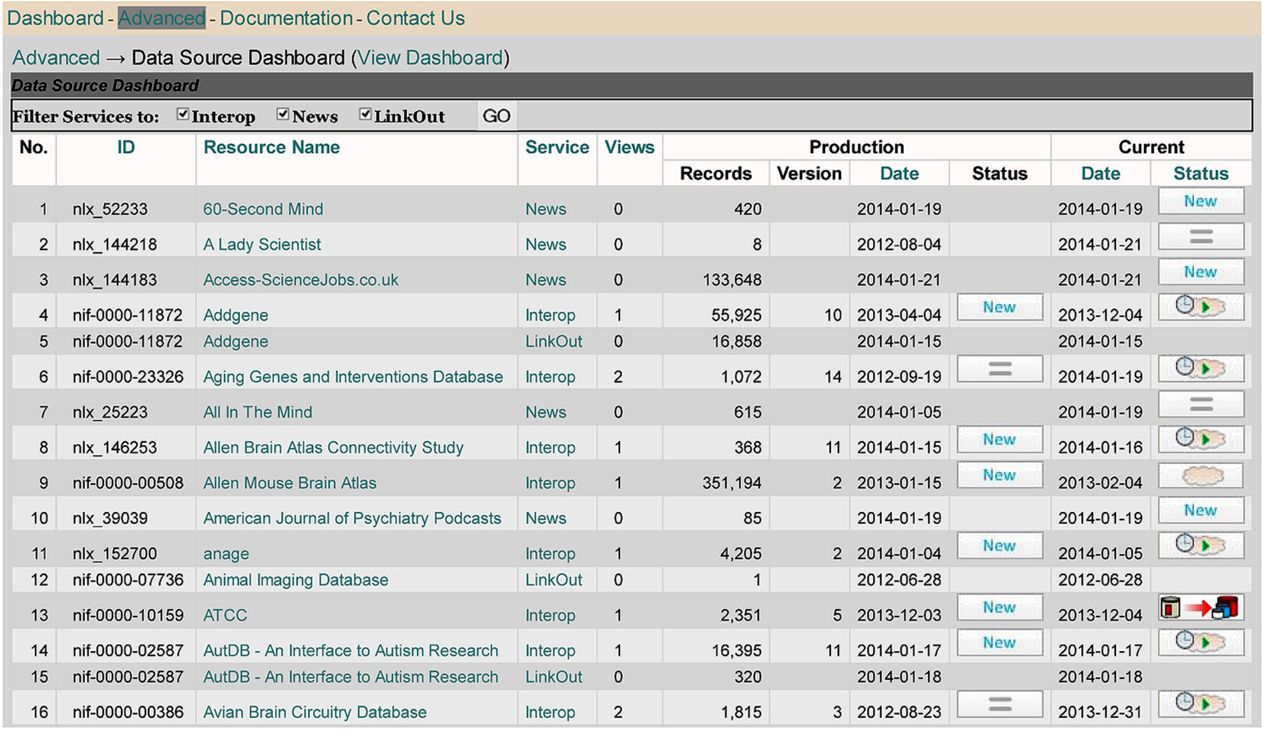
DISCO Information Edit

Current File:	Local (http://disco.neuinfo.org/discolocal/resource/nif-0000-00007/disco.xml)
Contact Name:	Willy WaiHo Wong
Contact Email:	wawong@sdsc.edu
Status:	parsed
Status Date:	2013-10-28 00:00:00
Local Version:	3.0
Remote Version:	1.1

DISCO Services

Type	URL	Format	Process schedule	Action	Schedule
Resource Description	http://disco.neuinfo.org/discolocal/resource/nif-0000-00007/disco.rd.xml	disco.rd.nif.rdf		View	
Terminology	http://ccdb.ucsd.edu/disco_lexicon.jsp	ndg.disco.lexicon.1			
Interoperation	http://ccdb.ucsd.edu/index.html	GIDI			
Interoperation	http://disco.neuinfo.org/discolocal/resource/nif-0000-00007/ccdb.xml	disco.interop		View	Monthly
Entrez LinkOut	http://disco.neuinfo.org/discolocal/resource/nif-0000-00007/linkoutsql.xml	disco.linkout.sql		View	Monthly
DB Schema	http://ccdb.ucsd.edu/disco_dbschema.jsp	ndg.disco.schema.1			

FIGURE 3 | A DISCO screen showing detailed information about an individual resource (Cell Centered Database) that participates in the NIF Federation.



Dashboard - Advanced - Documentation - Contact Us

Advanced → Data Source Dashboard (View Dashboard)

Data Source Dashboard

Filter Services to: ☑Interop ☑News ☑LinkOut GO

No.	ID	Resource Name	Service	Views	Production				Current	
					Records	Version	Date	Status	Date	Status
1	nlx_52233	60-Second Mind	News	0	420		2014-01-19		2014-01-19	New
2	nlx_144218	A Lady Scientist	News	0	8		2012-08-04		2014-01-21	=
3	nlx_144183	Access-ScienceJobs.co.uk	News	0	133,648		2014-01-21		2014-01-21	New
4	nif-0000-11872	Addgene	Interop	1	55,925	10	2013-04-04	New	2013-12-04	⏲▶
5	nif-0000-11872	Addgene	LinkOut	0	16,858		2014-01-15		2014-01-15	
6	nif-0000-23326	Aging Genes and Interventions Database	Interop	2	1,072	14	2012-09-19	=	2014-01-19	⏲▶
7	nlx_25223	All In The Mind	News	0	615		2014-01-05		2014-01-19	=
8	nlx_146253	Allen Brain Atlas Connectivity Study	Interop	1	368	11	2014-01-15	New	2014-01-16	⏲▶
9	nif-0000-00508	Allen Mouse Brain Atlas	Interop	1	351,194	2	2013-01-15	New	2013-02-04	☁
10	nlx_39039	American Journal of Psychiatry Podcasts	News	0	85		2014-01-19		2014-01-19	New
11	nlx_152700	anage	Interop	1	4,205	2	2014-01-04	New	2014-01-05	⏲▶
12	nif-0000-07736	Animal Imaging Database	LinkOut	0	1		2012-06-28		2012-06-28	
13	nif-0000-10159	ATCC	Interop	1	2,351	5	2013-12-03	New	2013-12-04	🗄→
14	nif-0000-02587	AutDB - An Interface to Autism Research	Interop	1	16,395	11	2014-01-17	New	2014-01-17	⏲▶
15	nif-0000-02587	AutDB - An Interface to Autism Research	LinkOut	0	320		2014-01-18		2014-01-18	
16	nif-0000-00386	Avian Brain Circuitry Database	Interop	2	1,815	3	2012-08-23	=	2013-12-31	⏲▶

FIGURE 4 | DISCO's NIF Data Source Dashboard.

to construct DISCO scripts that extract data from the resource.

- The second pipeline involves input data management. Input data management includes ingestion of data from each resource, validation, and version tracking. These processes are entirely managed within the DISCO system.
- The third pipeline involves output data management. This process includes NIF Federation dataset view generation and validation, including deployment of that data for use by the NIF community. DISCO coordinates with other components of the NIF to generate federated data Views to help make the data accessible to users in a flexible fashion.

DISCO utilizes customized template scripts that describe to the NIF crawling agent where data is located for each resource, how

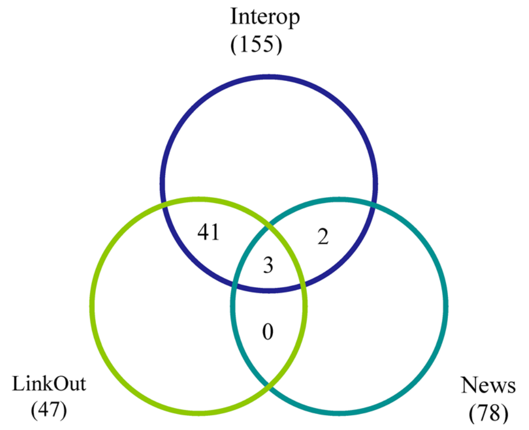


FIGURE 5 | A Venn diagram illustrating how the 231 resources supported by DISCO as of January 2014 are distributed among the three DISCO services (Interop, LinkOut, and News). The numbers indicate how many resources participate in each service and, of these, how many participate in two or three of the services.

it is to be extracted, and how it is to be stored in the NIF Federation warehouse. Source data in a variety of formats is supported, with new formats frequently incorporated. Within the warehouse, the resource data is stored using PostgreSQL tables. To facilitate provenance, tables from each resource are named using their NIF IDs as prefix. Every time a resource is rescanned, new temporary tables are generated. The new data is compared with the previous production version of the data for changes. If differences are found, the system reports a summary of the new changes to data curators for their verification. If the new data is accepted, DISCO checks whether there are any Views using this data. If so, new temporary Views are recreated and deployed to the NIF beta Website. Data curators as well as resource personnel are informed of a new version of these Views. Once the new temporary version of each View is approved, it is scheduled for production deployment. This process can be executed immediately or in batches, as desired.

DISCO was designed to deal with frequent changes in the information stored within a resource since such changes are common in neuroscience research. While most data changes consist of addition of new data or changes made to existing data, quite often a resource expands its content using new attributes or datasets. Less often the resource may reshape its contents using a different structure. Keeping track of these intra-resource domain changes over time is challenging unless these changes are properly documented within DISCO in a way that an automated agent can trace. DISCO scripts were particularly designed with this purpose in mind. Once a data extraction script is written, it is functional for data changes (additions, deletions, and corrections) as long as the structure of the data is not altered within the resource. If the structure changes, the extraction scripts need to be changed accordingly or the data ingestion procedure will break.

DISCO tracks data changes using predefined primary keys specified in resource scripts. (For a resource containing highly unstructured data, a hash of each entire record may be used.) These keys are used to create unique identifiers for specific pieces of data (entities) in a resource. The data structure within each resource is also tracked. DISCO uses a customized EAV/CR schema (Marenco et al., 2003) as a concurrent versioning system (CVS) back-end. Changes to data and metadata are stored using reverse delta methodology, and changes to resource database structures are stored using deltas. Reverse deltas allow DISCO to keep the most current version the data in the production tables actively used by the NIF, while changes from previous versions are stored in EAV form to allow the recreation of previous versions if needed. This technique is efficient for data additions and/or modest data edits. Substantial data changes may require copying all previous data to the CVS.

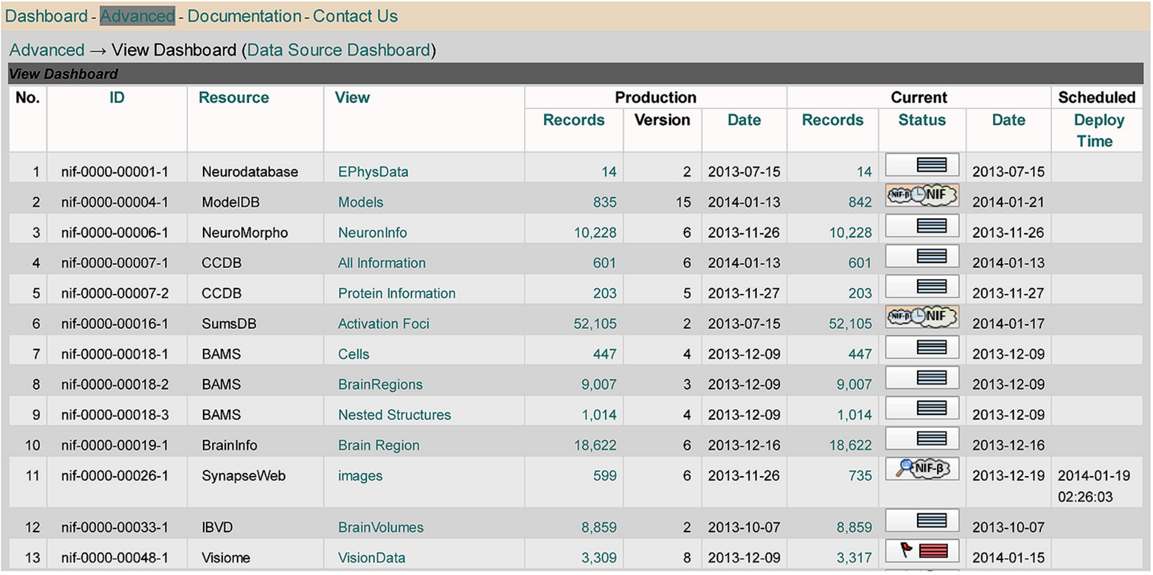
Semantic data mappings in DISCO are done based on schema annotations in DISCO scripts. We follow the approach to enhanced metadata annotation previously developed as the EAV/CR dataset protocol (EDSP; Marenco et al., 2003). Schema elements such as table groups, tables, and columns are annotated to describe their content, semantic relationships, and whether they contain complex objects, simple terms, or just values. Once data has been extracted from a resource, columns containing terms are queried to extract those terms, which are then mapped to term IDs from standard vocabularies. DISCO facilitates semantic data integration by mapping semantic metadata and term IDs to the NIFSTD ontology. The DISCO semantic data mapping functionality is coordinated by the NIF Concept Mapper (Gupta et al., 2008).

Having the most current data from each resource in DISCO is also challenging due to the absence of mechanisms to inform DISCO when new data is added to a resource. As described below, DISCO's current scheduling approach allows the data to be refreshed at predefined intervals. There is therefore no assurance of having completely up-to-date data from each resource. We are currently exploring mechanisms to allow bidirectional notifications from resources to prompt DISCO when to rescan for new information. For resources that are not able to implement this mechanism, an approach using some type of probing may be possible, for example, checking for file timestamps or size changes.

Since DISCO is a system in continual evolution, for its development we use the Scrum agile development framework. This allows quick development and rollup into production. DISCO is implemented as a Web application written in Java using PostgreSQL as a back-end database.

DISCO OPERATION

To coordinate its operation, DISCO contains three high-level "dashboards": the main DISCO Dashboard, the NIF Data Sources Dashboard, and the NIF Views Dashboard. These are used by NIF staff to coordinate the various workflow steps required to maintain the NIF Federation. The three dashboards provide an overview of the status of all of DISCO's various activities for every participating resource, along with the ability to drill down to see more detailed information about the activities and resources involved.



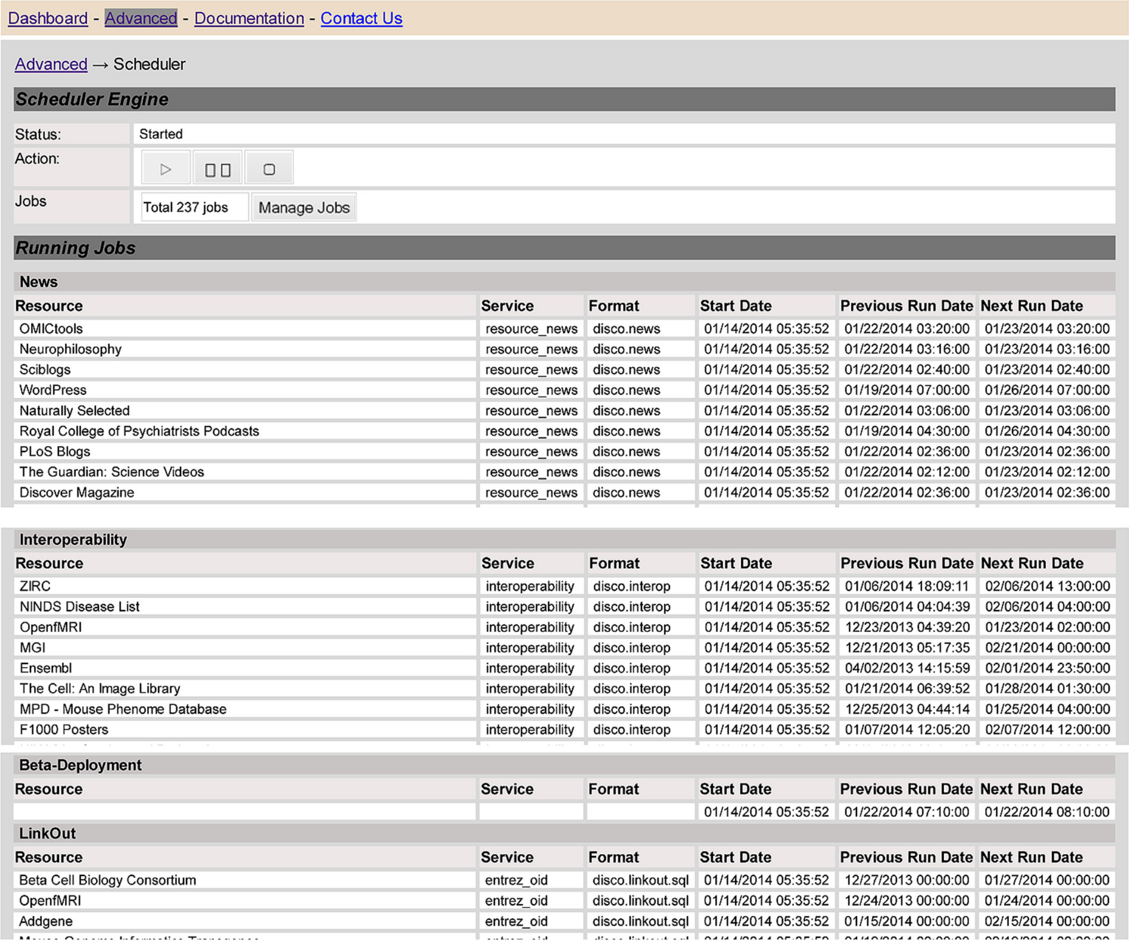
Dashboard - Advanced - Documentation - Contact Us

Advanced → View Dashboard (Data Source Dashboard)

View Dashboard

No.	ID	Resource	View	Production			Current			Scheduled
				Records	Version	Date	Records	Status	Date	Deploy Time
1	nif-0000-00001-1	Neurodatabase	EPhysData	14	2	2013-07-15	14	▤	2013-07-15	
2	nif-0000-00004-1	ModelDB	Models	835	15	2014-01-13	842	NIF-β NIF	2014-01-21	
3	nif-0000-00006-1	NeuroMorpho	NeuronInfo	10,228	6	2013-11-26	10,228	▤	2013-11-26	
4	nif-0000-00007-1	CCDB	All Information	601	6	2014-01-13	601	▤	2014-01-13	
5	nif-0000-00007-2	CCDB	Protein Information	203	5	2013-11-27	203	▤	2013-11-27	
6	nif-0000-00016-1	SumsDB	Activation Foci	52,105	2	2013-07-15	52,105	NIF-β NIF	2014-01-17	
7	nif-0000-00018-1	BAMS	Cells	447	4	2013-12-09	447	▤	2013-12-09	
8	nif-0000-00018-2	BAMS	BrainRegions	9,007	3	2013-12-09	9,007	▤	2013-12-09	
9	nif-0000-00018-3	BAMS	Nested Structures	1,014	4	2013-12-09	1,014	▤	2013-12-09	
10	nif-0000-00019-1	BrainInfo	Brain Region	18,622	6	2013-12-16	18,622	▤	2013-12-16	
11	nif-0000-00026-1	SynapseWeb	images	599	6	2013-11-26	735	NIF-β	2013-12-19	2014-01-19 02:26:03
12	nif-0000-00033-1	IBVD	BrainVolumes	8,859	2	2013-10-07	8,859	▤	2013-10-07	
13	nif-0000-00048-1	Visiome	VisionData	3,309	8	2013-12-09	3,317	⚑ ▤	2014-01-15	

FIGURE 6 | DISCO's NIF Views Dashboard.



Dashboard - Advanced - Documentation - Contact Us

Advanced → Scheduler

Scheduler Engine

Status:	Started
Action:	▷ ▯▯ ▢
Jobs	Total 237 jobs · Manage Jobs

Running Jobs

News

Resource	Service	Format	Start Date	Previous Run Date	Next Run Date
OMICtools	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 03:20:00	01/23/2014 03:20:00
Neurophilosophy	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 03:16:00	01/23/2014 03:16:00
Sciblogs	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 02:40:00	01/23/2014 02:40:00
WordPress	resource_news	disco.news	01/14/2014 05:35:52	01/19/2014 07:00:00	01/26/2014 07:00:00
Naturally Selected	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 03:06:00	01/23/2014 03:06:00
Royal College of Psychiatrists Podcasts	resource_news	disco.news	01/14/2014 05:35:52	01/19/2014 04:30:00	01/26/2014 04:30:00
PLoS Blogs	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 02:36:00	01/23/2014 02:36:00
The Guardian: Science Videos	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 02:12:00	01/23/2014 02:12:00
Discover Magazine	resource_news	disco.news	01/14/2014 05:35:52	01/22/2014 02:36:00	01/23/2014 02:36:00

Interoperability

Resource	Service	Format	Start Date	Previous Run Date	Next Run Date
ZIRC	interoperability	disco.interop	01/14/2014 05:35:52	01/06/2014 18:09:11	02/06/2014 13:00:00
NINDS Disease List	interoperability	disco.interop	01/14/2014 05:35:52	01/06/2014 04:04:39	02/06/2014 04:00:00
OpenfMRI	interoperability	disco.interop	01/14/2014 05:35:52	12/23/2013 04:39:20	01/23/2014 02:00:00
MGI	interoperability	disco.interop	01/14/2014 05:35:52	12/21/2013 05:17:35	02/21/2014 00:00:00
Ensembl	interoperability	disco.interop	01/14/2014 05:35:52	04/02/2013 14:15:59	02/01/2014 23:50:00
The Cell: An Image Library	interoperability	disco.interop	01/14/2014 05:35:52	01/21/2014 06:39:52	01/28/2014 01:30:00
MPD - Mouse Phenome Database	interoperability	disco.interop	01/14/2014 05:35:52	12/25/2013 04:44:14	01/25/2014 04:00:00
F1000 Posters	interoperability	disco.interop	01/14/2014 05:35:52	01/07/2014 12:05:20	02/07/2014 12:00:00

Beta-Deployment

Resource	Service	Format	Start Date	Previous Run Date	Next Run Date
			01/14/2014 05:35:52	01/22/2014 07:10:00	01/22/2014 08:10:00

LinkOut

Resource	Service	Format	Start Date	Previous Run Date	Next Run Date
Beta Cell Biology Consortium	entrez_oid	disco.linkout.sql	01/14/2014 05:35:52	12/27/2013 00:00:00	01/27/2014 00:00:00
OpenfMRI	entrez_oid	disco.linkout.sql	01/14/2014 05:35:52	12/24/2013 00:00:00	01/24/2014 00:00:00
Addgene	entrez_oid	disco.linkout.sql	01/14/2014 05:35:52	01/15/2014 00:00:00	02/15/2014 00:00:00

FIGURE 7 | The screen provides a snapshot overview that illustrates the operation of the DISCO task scheduler, as described in the text.

Figure 2 shows the main DISCO Dashboard, which also serves as DISCO's homepage. It contains the list of the resources, including all the NIF Federation resources, that share their information via DISCO. This dashboard (at the top of the "Resource" section) shows a toolbar for searching, sorting, and paging through the resources. Below is a table of resources, showing each resource's ID and name, as well as summary information indicating what NIF capabilities DISCO coordinates for that resource, as well as links to more detailed information. This information includes (1) where the DISCO file(s) for each resource resides (locally on the DISCO server or remotely at the resource itself), and (2) which DISCO services each resource is participating in. This dashboard provides NIF staff with an overview of this basic information about all participating resources.

Clicking on a resource name (e.g., Cell Centered Database in line 5 of **Figure 2**) will link to the DISCO content page for that resource, as shown in **Figure 3**. This page displays a variety of information about that resource, including contact information for that resource's technical support, as well as pointers to DISCO files which contain scripts defining in detail how DISCO implements each DISCO service for that resource (as summarized in the "Services" section of **Figure 2**). The content presented in this page is encoded using an XML formatted file, and can be modified by selecting the Edit button in the "DISCO Information" section of the page.

Whereas the main DISCO Dashboard provides an overview of the basic information DISCO maintains about each participating resource (including descriptive information and scripts), the NIF Data Source Dashboard (**Figure 4**) provides an overview of the *workflow status* of each resource.

The Data Source Dashboard displays a table with columns containing each resource's ID and name, together with the DISCO service type provided and summary information concerning the "Production" and "Current" version of the data that has been uploaded to the NIF for each service. DISCO currently supports three types of service. The basic service (labeled Interop) involves incorporating a set of specified data from a resource into the NIF Federation's data warehouse. The LinkOut service involves exporting data to the National Library of Medicine for incorporation in PubMed to support its ability to "link out" from a paper citation to related data items (Marenco et al., 2008). The News service allows DISCO to consolidate news provided by participating resources and to provide this aggregated news to interested NIF users. DISCO supports one or more of these three services for a total of 231 resources, as illustrated in **Figure 5**.

For example, as seen in **Figure 4**, the Addgene resource (lines 4 and 5) uses the DISCO Interop and LinkOut services. As indicated in **Figure 4**, the Addgene Interop data currently contains 55,925 records. The "Production" version of that data is in the 10th version of data uploaded, which was created on 4/4/13 and which involved the import of "New" data. In addition to this production version of the data, there is a more recent ("Current") version of the data that was uploaded on 12/4/13, which is "Pending" (as indicated by the little clock icon) inspection and approval by NIF staff before it can be used as the production version. When the [=] icon is shown in the status column (as seen in line 2 of **Figure 4**), this indicates that the most recent version of the data downloaded was unchanged from the previous version.

Underlying the information presented on this screen is a formalized NIF Data Source Lifecycle, which includes the following workflow.

- NIF staffs specify how frequently the data for each resource should be updated. This is determined by NIF staff in consultation with resource staff and depends on how frequently new data is added to a resource.
- When the time comes to update the data, a new version of the data is uploaded into a temporary table, where it is held (marked as "pending").
- The data is then compared against the production version of the data for that resource (unless of course this is the first version of data uploaded).
- If the data is unchanged from the production version, then that fact is recorded.
- If there is new data, and/or if previous data is changed, this fact is recorded, and the data continues to be held as "pending" until a NIF staff member reviews it (by inspecting the new data to assure that no errors or anomalies have occurred during the data import process).
- Based on this review, the NIF staff member may "approve" the newly uploaded version, in which case it will be scheduled for transfer to become the production version.
- If the NIF staff member identifies a potential problem, this fact is recorded. Depending on the nature of the problem a number of steps may take place next. Examples of the type of problems that occur when importing data include (1) data type errors, (2) duplicate keys, (3) text fields that are too big for the corresponding field within the NIF, and (4) failure of the data import process to complete.

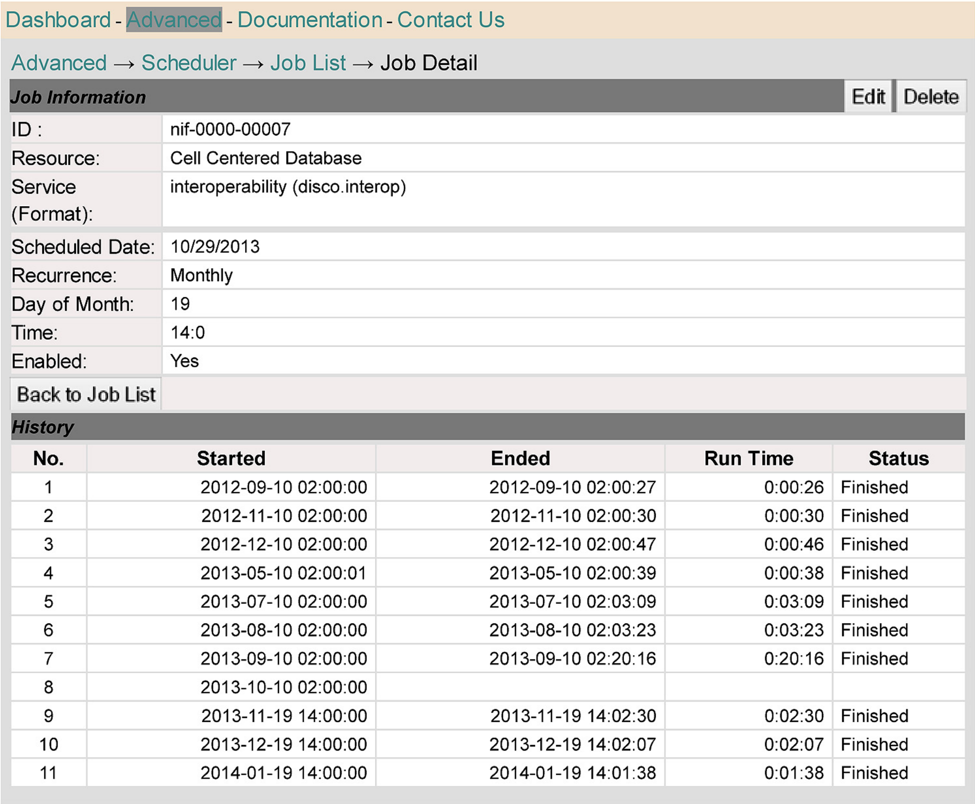
This coordination of the data sources lifecycle is the heart of DISCO's automated support of the workflow required to organize the ongoing harvest and integration of data from participating NIF Federation resources.

Figure 6 shows NIF Views Dashboard, which coordinates the maintenance of the various views that have been defined over the NIF Federation data, including a growing number of views that combine data elements from multiple resources. Views involving multiple resources are "materialized" in the sense that data elements from the NIF tables for each of those resources are copied into new table. This allows the combined data to be queried and manipulated more efficiently. The decision to create such a view is made by NIF staff in consultation with members of a community of neuroscience researchers for whom such a view would be helpful in presenting data in a fashion that would be most intelligible. See the online DISCO Manual[3] for more detail.

SCHEDULING AND COORDINATING THE DATA UPDATE TASKS

At any given point of time, different resources will be at different points within the overall data life cycle, and there may be many tasks that are waiting to be executed or in the process of

[3] http://disco.neuinfo.org/docs/manual/


Dashboard - Advanced - Documentation - Contact Us
Advanced → Scheduler → Job List → Job Detail
Edit
Delete
Back to Job List

Job Information

ID :	nif-0000-00007
Resource:	Cell Centered Database
Service (Format):	interoperability (disco.interop)
Scheduled Date:	10/29/2013
Recurrence:	Monthly
Day of Month:	19
Time:	14:0
Enabled:	Yes

History

No.	Started	Ended	Run Time	Status
1	2012-09-10 02:00:00	2012-09-10 02:00:27	0:00:26	Finished
2	2012-11-10 02:00:00	2012-11-10 02:00:30	0:00:30	Finished
3	2012-12-10 02:00:00	2012-12-10 02:00:47	0:00:46	Finished
4	2013-05-10 02:00:01	2013-05-10 02:00:39	0:00:38	Finished
5	2013-07-10 02:00:00	2013-07-10 02:03:09	0:03:09	Finished
6	2013-08-10 02:00:00	2013-08-10 02:03:23	0:03:23	Finished
7	2013-09-10 02:00:00	2013-09-10 02:20:16	0:20:16	Finished
8	2013-10-10 02:00:00			
9	2013-11-19 14:00:00	2013-11-19 14:02:30	0:02:30	Finished
10	2013-12-19 14:00:00	2013-12-19 14:02:07	0:02:07	Finished
11	2014-01-19 14:00:00	2014-01-19 14:01:38	0:01:38	Finished

FIGURE 8 | This screen shows scheduling information relevant to a specific resource, the Cell Centered Database.

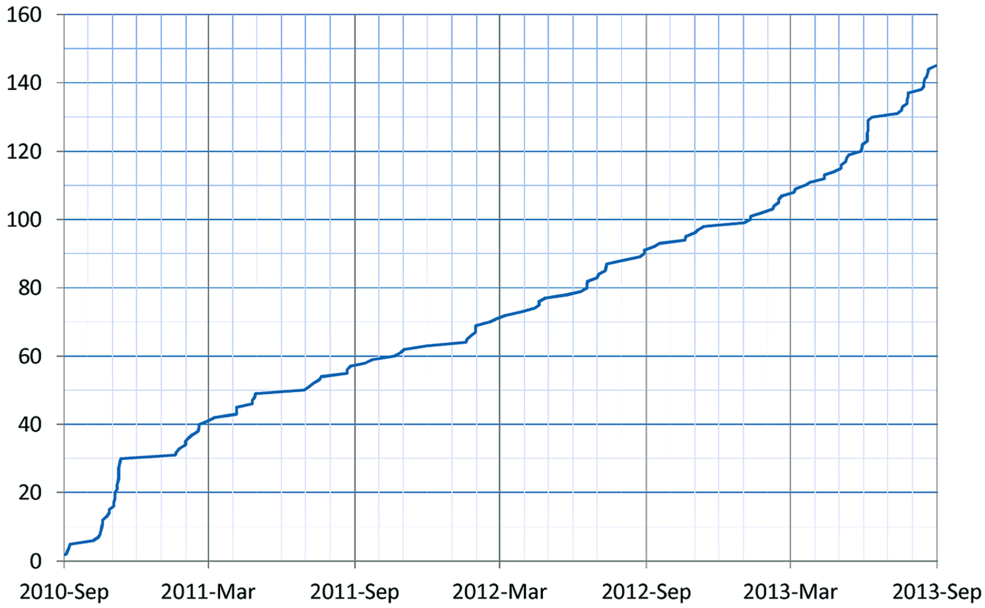


FIGURE 9 | Growth of the NIF Federation in terms of the number of participating resources over time.

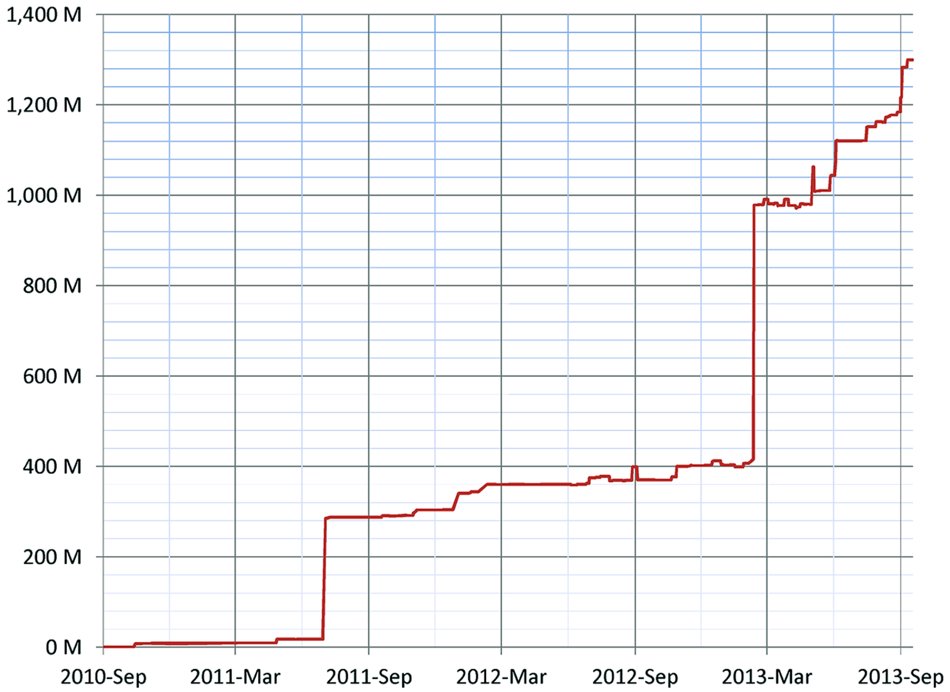


FIGURE 10 | Growth of the NIF Federation in terms of the number of records (the number of rows of data) stored over time. There are two major jumps in seen in the graph: (1) in July 2011 BrainSpan.org (an atlas of the development of human brain) was added with 267 million records, and (2) in February 2013 PubMed was added with 567 million records.

Table 1 | This table shows how frequently the data from different Interop resources are updated within the NIF Federation, as of January 2014.

	# of resources
Weekly	12
Bi-weekly	4
Monthly	122
Ad hoc (not scheduled)	17
Total	155

being executed. DISCO has a number of components to help manage, schedule and coordinate all these activities. To illustrate how DISCO manages these activities, **Figure 7** shows a Web page that provides an overview "snapshot" of the activities of the DISCO scheduler engine as of a given point of time. This table lists the various resources that are scheduled to be updated, or are in the process of being updated.

Figure 8 provides a different perspective on the scheduling function supported by DISCO. In this case, we see the process from the perspective of a single resource, in this case the Cell Centered Database. As indicated in the top half of the screen, the data for this resource is updated on a monthly basis, currently on the 19th of each month, at 2 P.M. The bottom half of the screen shows a record of the six most recent update runs.

This section has provided an overview of DISCO's activities by showing a representative subset of the various screens that DISCO provides to help manage and coordinate the integration of data within the NIF Federation. Our goal in showing and describing these screens has been to help make the various functions that DISCO provides more concrete and transparent.

CURRENT STATUS AND FUTURE DIRECTIONS

As of January 2014, 155 resources utilize the DISCO Interop service to share data via the NIF Federation. **Figure 9** shows how this number has gradually increased over time. The relatively steady rate of increase reflects the fact that the amount of effort to incorporate a new resource is relatively constant irrespective of the amount of data involved. **Figure 10** shows how the amount of data stored in the NIF Federation has increased over the same time period. **Table 1** indicates how frequently the NIF Federation resources are currently updated. **Figure 11** illustrates this process from the perspective of a single participating resource by showing how the amount of data stored in the NIF Federation for ModelDB has grown over time.

Looking to the future, the development of DISCO will continue to be a work in progress. The overall approach is undergoing a continual process of refinement. There is also a quite extensive list of additional capabilities that would be desirable to incorporate in the future.

- There are a number of ways in which the current DISCO system could be refined and made more robust. For example, there

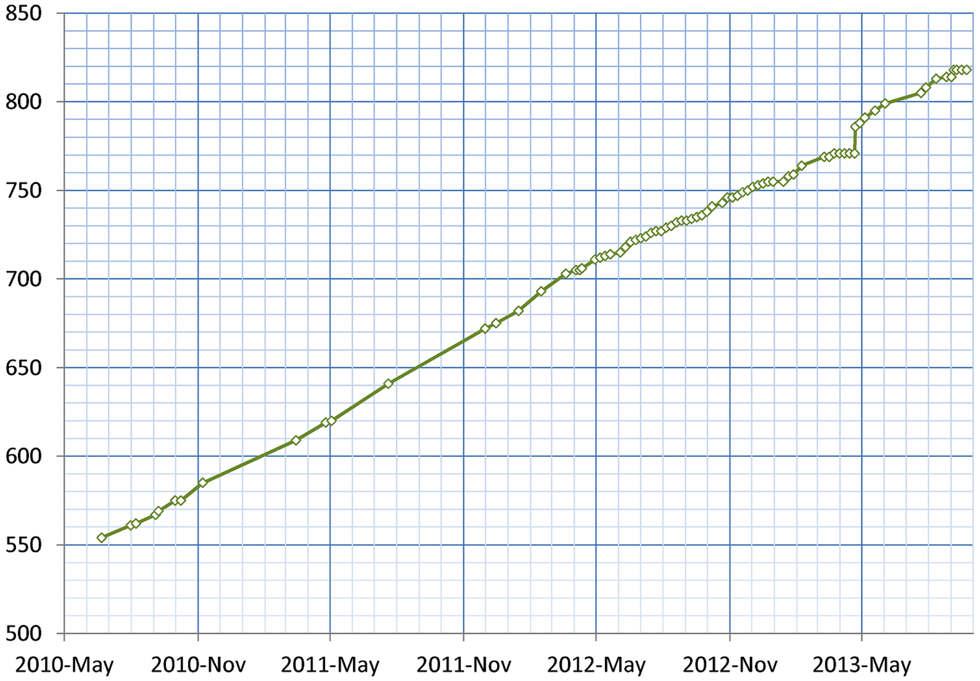


FIGURE 11 | This figures shows the growth of data from ModelDB stored in the NIF Federation, reflecting the growth of ModelDB itself, and the ongoing process of harvesting that data and integrating it into the NIF Federation. Each diamond indicates a time when data was updated.

is a need for more extensive status reporting and debugging tools for use when the import of data from a resource "hangs" (fails to complete). There are a wide variety of reasons this might happen, and it does happen quite regularly. It is a major problem that needs to be accommodated by providing as much automated assistance as possible.

- A second refinement that will be important as the NIF Federation grows will be to distribute DISCO's functions over multiple machines so that the many tasks that are performed can be accomplished more rapidly utilizing parallel computing. DISCO currently runs on a single server machine.
- In addition to enhancing the current DISCO framework, there is a wide range of further capabilities that we would like to build. A major project would be to integrate the NIF's ontology mapping functions so that they can be applied in an automated fashion to data as they are being imported. It is also becoming evident that the underlying capabilities implemented in DISCO could be utilized in other data aggregator systems, and that other groups would like to leverage the DISCO code. Some of these will want to extract data from some of the same resources as DISCO. The DISCO code could be adapted to facilitate its use by other groups including the shared harvesting of data from a resource by multiple aggregator systems.

DISCUSSION

This section discusses some of the challenges that face the various groups of people who participate in developing and maintaining NIF and DISCO, including (1) the DISCO developers, (2) the NIF staff who use DISCO to coordinate their activities, and (3) the local staff at the participating resources. The biggest challenge facing all of these groups is that Web-based resources can be very idiosyncratic and difficult to extract data from, for a variety of reasons.

LACK OF DESCRIPTIVE METADATA

One example of this problem is seen when a table containing data to be downloaded into the NIF does not include descriptive metadata such as informative column headers (e.g., a table might contain several idiosyncratic column names ("str1,""str2,"...) without any metadata indicating that these columns contain the names of "strains") or descriptive data types (e.g., a table might contain "1" as a data type instead of "male"). When descriptive, informative metadata is used in the tables downloaded from a resource, that metadata can in turn be used to more effectively annotate the data within the NIF to facilitate searching that data and integrating it with other data sets.

USE OF COMPLEX DATA-PRESENTATION LOGIC

Another problem is seen when resource developers use customized client-side code (such as JavaScript) to present data in their local Web site. This approach has the advantage of allowing the Web presentation of the data to be "flashier" and potentially more understandable by the resource's users. Unfortunately for DISCO staff, the approach also results in the data being in effect concealed by the client-side programming. To extract data from such a Web site, DISCO staff must examine and understand this code in detail to understand exactly what it is doing, so that they can write appropriate logic to access the data.

- A simple approach that greatly facilitates data extraction from a Web-based resource is for the resource developers to use a standardized template to organize the data presented on each Web page. For example, a standard set of section headers that are located consistently within each Web page greatly facilitates the organized extraction of that data. In addition, using standard terms for column headers greatly facilitates semi-automated terminology mapping of resource terms by the NIF Concept Mapper (Gupta et al., 2008).

It will be important to develop guidelines and standards for resource Web-site design that can be used by a resource to facilitate incorporation into a system like DISCO. New Web standards such as HTLM5, RDFa, and Google Microformats are very productive steps in this direction since they encourage and facilitate the incorporation of semantic metadata describing a resource's data. The increasing use of these approaches in the design of Web resources will facilitate automated data extraction by data aggregators such as DISCO.

INFORMATION SHARING STATEMENT

Technical information describing DISCO, including installation instructions, is available at: http://disco.neuinfo.org/docs/manual/. Additional program code can be obtained by contacting project staff.

ACKNOWLEDGMENTS

This research was supported in part by NIH grants R01 DC009977 and R01 DA021253, by NIH Neuroscience Blueprint Contract HHSN271200577531C from NIDA, and by the Society for Neuroscience.

REFERENCES

Bandrowski, A. E., Cachat, J., Li, Y., Müller, H. M., Sternberg, P. W., Ciccarese, P., et al. (2012). A hybrid human and machine resource curation pipeline for the Neuroscience Information Framework. *Database* (*Oxford*) 2012:bas005. doi: 10.1093/database/bas005

Bug, W. J., Ascoli, G. A., Grethe, J. S., Gupta, A., Fennema-Notestine, C., Laird, A. R., et al. (2008). The NIFSTD and BIRNLex vocabularies: building comprehensive ontologies for neuroscience. *Neuroinformatics* 6, 175–194. doi: 10.1007/s12021-008-9032-z

Cachat, J., Bandrowski, A., Grethe, J. S., Gupta, A., Astakho, V., Imam, F., et al. (2012). A survey of the neuroscience resource landscape: perspectives from the neuroscience information framework. *Int. Rev. Neurobiol.* 103, 39–68. doi: 10.1016/B978-0-12-388408-4.00003-4

Fischer, M., Thai, Q. K., Grieb, M., and Pleiss, J. (2006). DWARF – a data warehouse system for analyzing protein families. *BMC Bioinformatics* 7:495. doi: 10.1186/1471-2105-7-495

Gardner, D., Akil, H., Ascoli, G. A., Bowden, D. M., Bug, W., Donohue, D. E., et al. (2008). The Neuroscience Information Framework: a data and knowledge environment for neuroscience. *Neuroinformatics* 6, 149–160. doi: 10.1007/s12021-008-9024-z

Gupta, A., Bug, W., Marenco, L., Qian, X., Condit, C., Rangarajan, A., et al. (2008). Federated access to heterogeneous information resources in the Neuroscience Information Framework (NIF). *Neuroinformatics* 6, 205–217. doi: 10.1007/s12021-008-9033-y

Imam, F. T., Larson, S. D., Bandrowski, A., Grethe, J. S., Gupta, A., and Martone, M. E. (2012). Development and use of Ontologies Inside the Neuroscience Information Framework: a Practical Approach. *Front. Genet.* 3:111. doi: 10.3389/fgene.2012.00111

Jones, P., Binns, D., McMenamin, C., McAnulla, C., and Hunter, S. (2011). The InterPro BioMart: federated query and web service access to the InterPro Resource. *Database* (*Oxford*) 2011:bar033. doi: 10.1093/database/bar033

Majeed, R. W., and Röhrig, R. (2012). Automated realtime data import for the i2b2 clinical data warehouse: introducing the HL7 ETL cell. *Stud. Health Technol. Inform.* 180, 270–274.

Marenco, L., Giorgio, A., Ascoli, G. A., Martone, M. E., Shepherd, G. M., and Miller, P. L. (2008). The NIF LinkOut Broker: a web resource to facilitate federated data integration using NCBI identifiers. *Neuroinformatics* 6, 219–227. doi: 10.1007/s12021-008-9025-y

Marenco, L., Tosches, N., Crasto, C., Shepherd, G., Miller, P. L., and Nadkarni, P. M. (2003). Achieving evolvable Web-database bioscience applications using the EAV/CR framework: recent advances. *J. Am. Med. Inform. Assoc.* 10, 444–453. doi: 10.1197/jamia.M1303

Marenco, L., Wang, R., Shepherd, G. M., and Miller, P. L. (2010). The NIF DISCO Framework: facilitating automated integration of neuroscience content on the web. *Neuroinformatics* 8, 101–112. doi: 10.1007/s12021-010-9068-8

Saltz, J., Oster, S., Hastings, S., Langella, S., Kurc, T., Sanchez, W., et al. (2006). caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid. *Bioinformatics* 22, 1910–1916. doi: 10.1093/bioinformatics/btl272

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 22 January 2014; accepted: 06 May 2014; published online: 28 May 2014.

Citation: Marenco LN, Wang R, Bandrowski AE, Grethe JS, Shepherd GM and Miller PL (2014) Extending the NIF DISCO framework to automate complex workflow: coordinating the harvest and integration of data from diverse neuroscience information resources. Front. Neuroinform. ***8****:58. doi: 10.3389/fninf.2014.00058*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 06 January 2014
doi: 10.3389/fninf.2013.00049

TheHiveDB image data management and analysis framework

J-Sebastian Muehlboeck [1,2,3] *, **Eric Westman** [1,2] † **and Andrew Simmons** [1,4,5] †

[1] Department of Neuroimaging, Institute of Psychiatry, King's College London, London, UK
[2] Department of Neurobiology, Care Sciences and Society, Karolinska Institutet, Stockholm, Sweden
[3] J-S Muehlboeck Inc., Montreal, QC, Canada
[4] NIHR Biomedical Research Centre for Mental Health, King's College London, London, UK
[5] NIHR Biomedical Research Unit for Dementia, King's College London, London, UK

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Kevin Archie, Washington University School of Medicine, USA
Samir Das, Montreal Neurological Institute, Canada
Jihad S. Obeid, Medical University of South Carolina, USA

***Correspondence:**
J-Sebastian Muehlboeck, J-S Muehlboeck Inc., 20-7945 Rue Nadeau, Brossard QC J4Y 0G7, Canada
e-mail: muehlboeck@gmail.com

† *Eric Westman and Andrew Simmons are joint last authors.*

The hive database system (theHiveDB) is a web-based brain imaging database, collaboration, and activity system which has been designed as an imaging workflow management system capable of handling cross-sectional and longitudinal multi-center studies. It can be used to organize and integrate existing data from heterogeneous projects as well as data from ongoing studies. It has been conceived to guide and assist the researcher throughout the entire research process, integrating all relevant types of data across modalities (e.g., brain imaging, clinical, and genetic data). TheHiveDB is a modern activity and resource management system capable of scheduling image processing on both private compute resources and the cloud. The activity component supports common image archival and management tasks as well as established pipeline processing (e.g., Freesurfer for extraction of scalar measures from magnetic resonance images). Furthermore, via theHiveDB activity system algorithm developers may grant access to virtual machines hosting versioned releases of their tools to collaborators and the imaging community. The application of theHiveDB is illustrated with a brief use case based on organizing, processing, and analyzing data from the publically available Alzheimer Disease Neuroimaging Initiative.

Keywords: neuroimaging database framework, image processing, query interface, data management, data query, neuroimaging collaboration and workflows, web 2.0 application

INTRODUCTION

The advent of increasing numbers of large longitudinal imaging studies, imaging-genetics studies, and multi-center studies and the need to curate large volumes of imaging data from individual studies for data reuse purposes has led to a growing need for an integrated brain imaging database, resource, data, and activity management system. A number of imaging databases have been described in the literature including the LONI IDA (Van Horn and Toga, 2009), Loris (Das et al., 2012), and XNAT (Marcus et al., 2007) systems. Each of these databases represent attempts to create a system capable of jointly managing the increasing amounts of imaging data and data from other sources and modalities, while providing support for the specific processing requirements of imaging projects. They have been created in and for very specific environments with their own respective emphases and limitations.

The driver for the creation of a new alternative approach arose from a series of joint studies between King's College London, the Karolinska Institute, and our collaborators working on a number of large imaging studies including AddNeuroMed (Lovestone et al., 2007, 2009), Alzheimer Disease Neuroimaging Initiative (ADNI; Jack et al., 2008; Weiner et al., 2010), and AIBL (Ellis et al., 2009). The hive database system (theHiveDB) has been developed to match requirements not easily reconciled with the alternatives mentioned above. TheHiveDB offers a consistent solution to the intricacies of imaging projects. For ongoing projects and pre-existing collections of data it provides viable approaches to properly organize, manage, and store, both imaging and associated non-imaging data types. It is first and foremost a data aggregation and management system with a focus on easy interactions with the researcher.

MATERIALS AND METHODS

NEUROIMAGING PROJECT CHARACTERISTICS

Imaging projects consist of sets of participants, referred to here as individuals, typically divided into different groups (e.g., patients and healthy controls, or those who respond/don't respond to the effects of a novel drug). According to study protocols individuals may present on a number of occasions. These might be visits for cognitive tests or scanning sessions. Once data is acquired it is assigned to predefined labels (e.g., Baseline, 1-year-follow-up etc.), referred to here as timepoints. Imaging data is acquired in conjunction with a plethora of clinical, behavioral, and genetic data. Data from these modalities are frequently available in tabular form and often need to be combined across modalities for subsequent analysis. To properly support imaging data a neuroimaging database framework also needs to support the management of binary files, which we refer to as assets. We will consider here a use case of magnetic resonance imaging data, though the system is designed flexibly so that PET, SPECT, digital X-rays, or other medical images can also be managed.

At each timepoint a study consists of a series of images (for example a MRI localizer, multi-slice T2-weighted fast spin echo images and a T1-weighted ultrafast gradient echo volume). Each individual series will often consist of a number of slices or volumes. To guarantee usable and comparable results, scanning protocols are often pre-defined and matched to other imaging studies. TheHiveDB is designed to manage and organize raw and processed imaging data in conjunction with other available data such as demographic, cognitive, biological sample, and genetic data. Special attention and support are given to raw imaging data which is efficiently archived, properly stored and thoroughly documented. Image types can be defined by means of scanning protocol parameters, such that image assets can be extracted automatically from raw data archives. The system provides image format conversion routines for the resulting image assets. Assets are accessible to authorized users (project members) through a web interface via secure streaming. Tabular data can be downloaded through an interactive query interface.

For image processing, an activity component allows the execution and automation of frequent imaging tasks or application of standard image processing pipelines by means of a convenient web interface. Activities are defined in terms of the required inputs and resources designated to carry them out. Activity instances can be created by resource owners and assigned to projects.

For effective network management and security the authentication, authorization, and accounting (AAA) architecture has been chosen. System users need to authenticate to access the system. The authorization function is split into access to file data (assets), which is granted by means of project memberships (and is possible via the web interface) and user roles. The latter define the extent to which a user can interact with the system (e.g., only query data versus upload data and request processing). The activity system provides tracking and accounting functionality.

The main aspects of the system are:

- Asset management – storage, data archival, retrieval/access, availability, transfer, backup.
- Data processing – rendering algorithms available and usable for projects in an automated and traceable fashion.
- Resource management and sharing – to reduce overhead and cost, existing resources can be managed effectively and shared efficiently.
- Data querying – interactive querying of variables of interest across modalities.

APPLICATION ARCHITECTURE

The Hive database web application has been developed using the Grails open source web application framework based on the Groovy programming language. Groovy is an object-oriented programming language for the Java platform, which is dynamically compiled to java virtual machine (JVM) byte-code. Since most Java code is also syntactically valid Groovy code it interoperates seamlessly with existing Java code and libraries. The Grails framework interacts with relational database engines using object relational mapping. Hibernate[1] is used for relational persistence. MySQL has been chosen as the default database for theHiveDB due to its performance, wide-spread availability, transactional support, and web and data warehouse strengths[2]. As a full stack web application framework Grails provides performance optimized layers for communication with the back end, domain object mapping, database communication, and caching. Our current production environment hosts imaging for about 18,000 scanning sessions with 50,000 series and over 33 million documented DICOM files. The entire application is connected to various profiling utilities to identify and address scenarios where response times for the web interface are above 800 ms.

TheHiveDB relies on job scheduling[3] for any request likely to use significant CPU resources (e.g., run Freesurfer or DICOM archive creation). For these requests a job record is created with instant feedback to the user. The same applies for data transfers. The system handles jobs and transfers independently of the user's session based on resource availability, priorities, and concurrent requests.

The web application interface is accessible via secure http (https), which provides bidirectional encryption of communications between client and server. The system communicates with all resources using a pure Java implementation of the SSH-2 protocol[4]. The web interface relies heavily on JavaScript libraries to enhance the user's experience. JavaScript libraries are used within the context specific help system, data filters (see **Figure 1**), and the dynamic query interface. Additionally some views have JavaScript enhancements to allow for viewing adjustments.

The activity system extensively uses the open source grid engine [formerly Sun grid engine (sge)] for job scheduling, monitoring and resource management. Grid Engine is software that facilitates "distributed resource management" (DRM). Far more than just simple load-balancing tools or batch scheduling mechanisms, DRM software typically provides the following key features across large sets of distributed resources[5]:

- Policy based allocation of distributed resources (CPU time, software licenses, etc.)
- Batch queuing and scheduling
- Supports diverse server hardware, operating systems (OSs), and architectures
- Load balancing and remote job execution
- Detailed job accounting statistics
- Fine-grained user specifiable resources
- Suspension, resumption, and migration of jobs
- Tools for reporting Job/Host/Cluster status
- Job arrays
- Integration and control of parallel jobs

The integration of other job schedulers within theHiveDB is feasible as long as they support the features listed above.

STORAGE ARCHITECTURE

TheHiveDB facilitates the work of research groups by offering a unified approach to management, sharing, and processing of imaging data research projects. It has been designed as an imaging

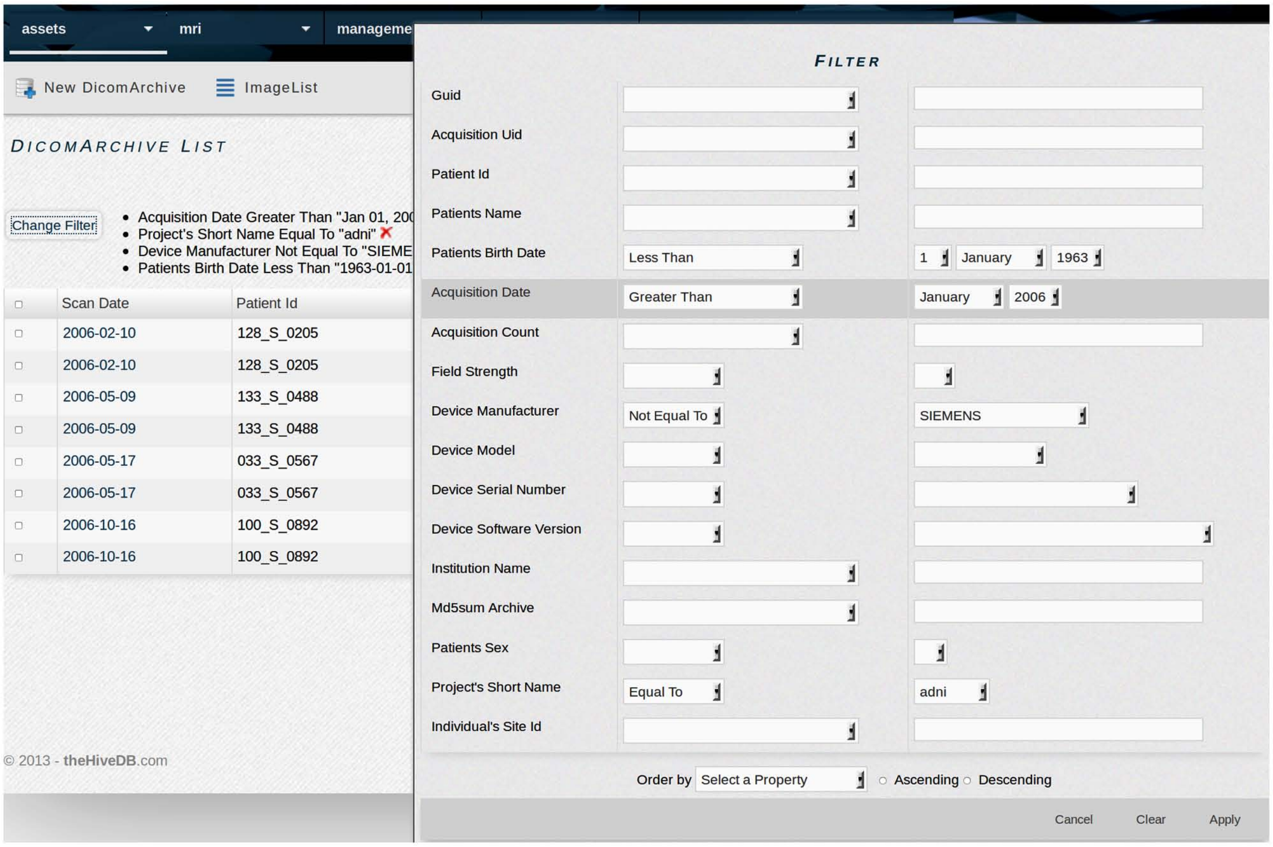
[1] http://www.hibernate.org/

[2] http://www.mysql.com/why-mysql/topreasons.html

[3] http://quartz-scheduler.org/

[4] http://www.snailbook.com/protocols.html

[5] http://packages.debian.org/wheezy/gridengine-client



DICOMARCHIVE LIST

- Acquisition Date Greater Than "Jan 01, 200
- Project's Short Name Equal To "adni" ✗
- Device Manufacturer Not Equal To "SIEME
- Patients Birth Date Less Than "1963-01-01

□	Scan Date	Patient Id
□	2006-02-10	128_S_0205
□	2006-02-10	128_S_0205
□	2006-05-09	133_S_0488
□	2006-05-09	133_S_0488
□	2006-05-17	033_S_0567
□	2006-05-17	033_S_0567
□	2006-10-16	100_S_0892
□	2006-10-16	100_S_0892

© 2013 - theHiveDB.com

FILTER

Guid		
Acquisition Uid		
Patient Id		
Patients Name		
Patients Birth Date	Less Than	1 January 1963
Acquisition Date	Greater Than	January 2006
Acquisition Count		
Field Strength		
Device Manufacturer	Not Equal To	SIEMENS
Device Model		
Device Serial Number		
Device Software Version		
Institution Name		
Md5sum Archive		
Patients Sex		
Project's Short Name	Equal To	adni
Individual's Site Id		

Order by Select a Property ○ Ascending ○ Descending

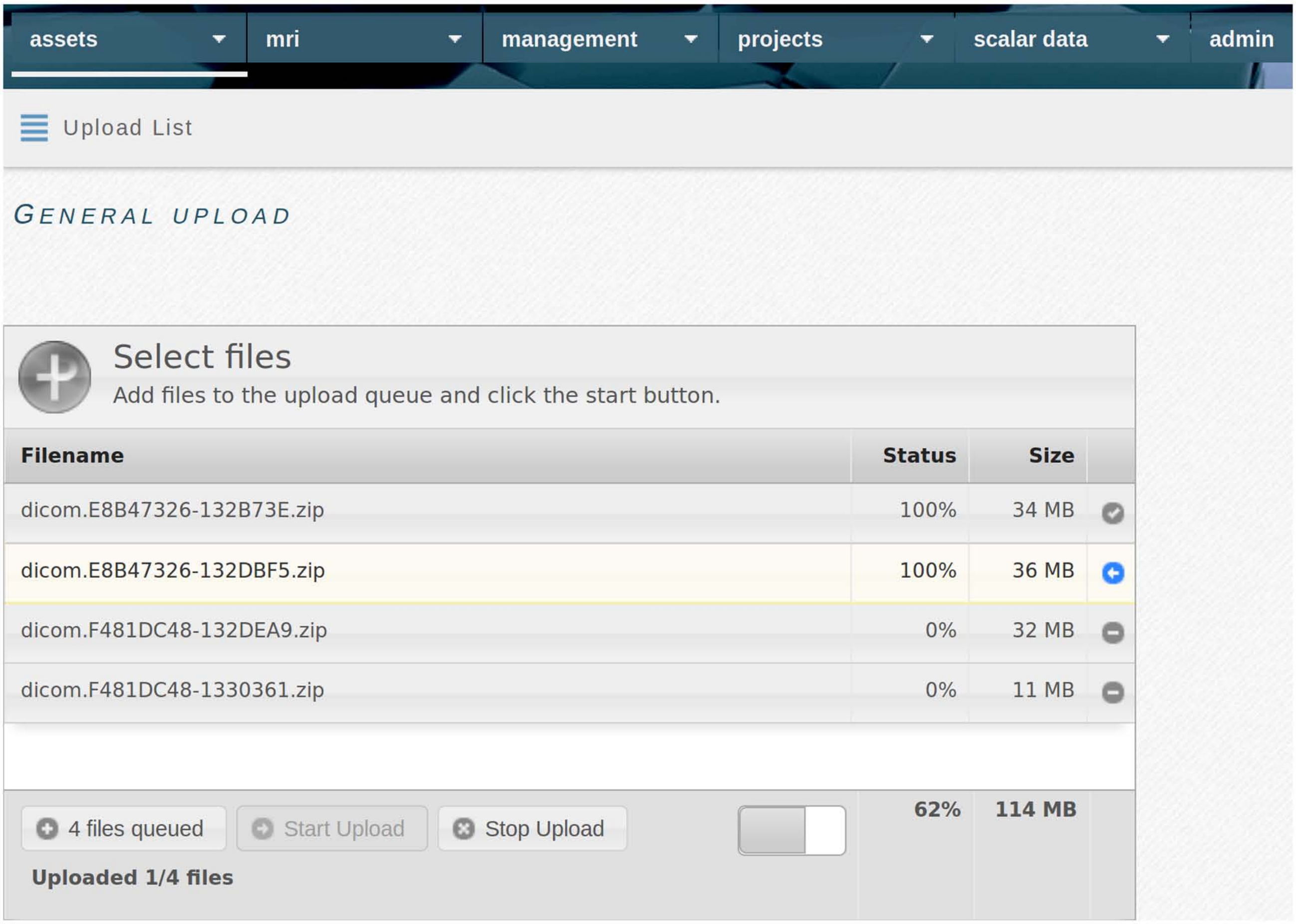
FIGURE 1 | TheHiveDB provides extensive filters for searching within entity lists (e.g., individuals or different types of assets). The example shows a filter used for searching ADNI DICOM archive data using specific criteria like data acquired for individuals born prior to 1963 and scanned after January 2006 on a non-Siemens scanner.

project and data management system with an integrated activity component. Imaging projects are created using the web interface. Study participants (individuals) are assigned to projects using project specific identifiers. Individuals can be created and maintained through the web interface, direct upload of individual lists or automatically derived from DICOM[6] header data.

All file data enters the system through a web-based upload interface (**Figure 2**). File naming conventions and manual assignment can be used for allocation to projects. Uploaded tabular data is incorporated directly (e.g., individual list or cognitive test result; see **Figure 3**), while (binary) files are recorded as assets. Assets are data entities managed by theHiveDB. They are registered upon creation or upload and can be transferred for processing or downloaded via streaming through the web interface. Every asset belongs to a project, individual, and timepoint by virtue of being assigned to it directly (e.g., an image) or by inheritance (e.g., an image transform, the modified representation of an image outputted by an image processing algorithm).

To manage assets effectively theHiveDB relies on predictable unique identifiers. TheHiveDB automatically computes and assigns such identifiers to all newly created assets. The identifiers are predictable, because they are determined based on information about the actual asset or the process leading to its creation. Technically the identifier is a deterministic universally unique identifier (dUUID). A UUID is a 16-octet (128-bit) number. In canonical form, it is represented by 32 hexadecimal digits, displayed in five groups separated by hyphens for a total of 36 characters (8-4-4-4-12, i.e., 32 alphanumeric characters and four hyphens, e.g., 6d0b1c00-2a11-4aaa-a337-3ba06e9ee2ef). UUIDs are frequently used in distributed systems to uniquely identify information. A UUID by itself is not human interpretable. Within theHiveDB however it is used as a powerful alias for the asset it refers to. TheHiveDB web interface offers the possibility to use UUIDs like tracking numbers and will assemble details for all assets listed in the search field. User preferences govern how assets are renamed for the individual user upon download. If the above example for instance refers to a DICOM archive, the user may choose to retrieve such files as managed by the system (i.e., 6d0b1c00-2a11-4aaa-a337-3ba06e9ee2ef.tar), identified for asset type (i.e., dicomArchive.6d0b1c00-2a11-4aaa-a337-3ba06e9ee2ef.tar), enriched with human-interpretable information (e.g., DCM.AcquisitionDate.PatientID.6d0b1c00-2a11-4aaa-a337-3ba06e9ee2ef.tar), etc. Similar renaming options are available for other asset types.

[6] http://medical.nema.org/standard.html



Add files to the upload queue and click the start button.

Filename	Status	Size
dicom.E8B47326-132B73E.zip	100%	34 MB
dicom.E8B47326-132DBF5.zip	100%	36 MB
dicom.F481DC48-132DEA9.zip	0%	32 MB
dicom.F481DC48-1330361.zip	0%	11 MB
	62%	114 MB

Uploaded 1/4 files

FIGURE 2 | TheHiveDB features a web based upload interface, which allows local data to be uploaded to the database. The multi file upload allows for drag and drop and shows upload progress.

Aside from warranting uniqueness, predictability is another concern. Therefore within theHiveDB UUIDs are not assigned randomly, but computed in a deterministic fashion. For instance DICOM header information is used to compute identifiers for image assets. Identifiers for output from image processing algorithms or pipelines are computed taking the algorithm's name, version, and input file identifiers into account. Consequently, requesting extraction of images from a DICOM archive containing a subset of already extracted data will result in a UUID collision. Similarly, the request to reprocess data with the same algorithm without removing previous results will fail. While there is currently no plan to implement federated searches, data exchange, or migration between HiveDB instances is planned.

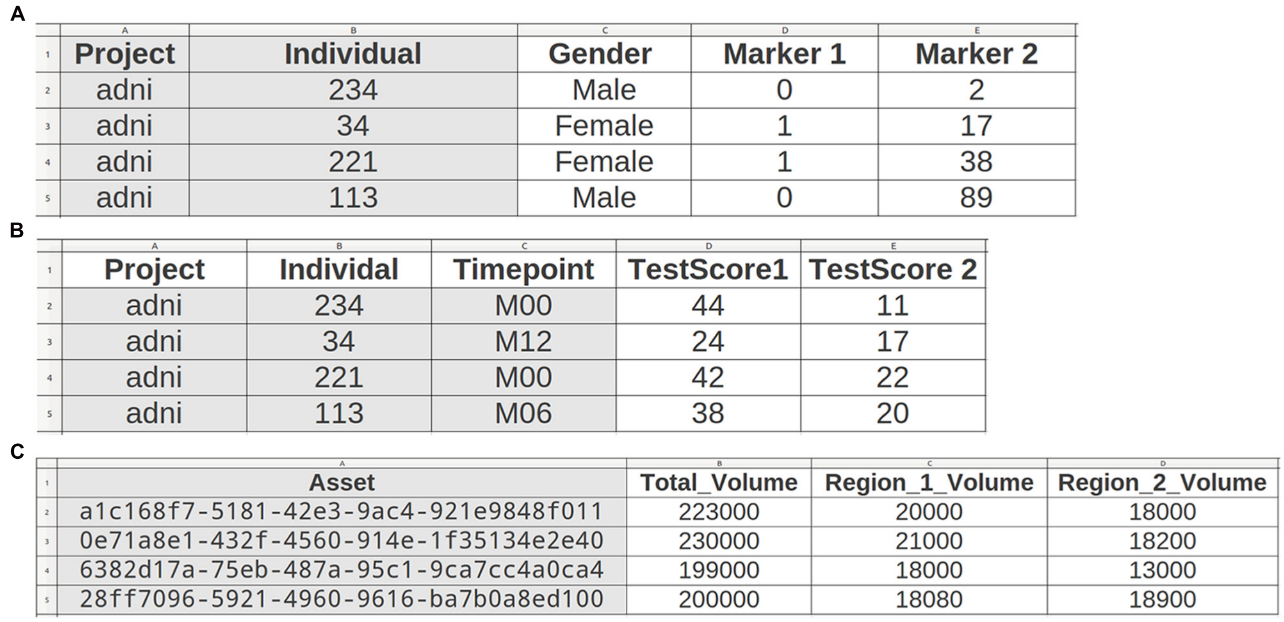
Typically assets will have at least one "asset file" – the data file on disk associated with it. These asset files may exist at multiple locations (e.g., one in project space and another one as backup in the cloud).

Being tailored to the specific needs of brain imaging projects the system extends the notion of asset to a number of special assets like DICOM archives (see section "DICOM management, storage, and compression"), images (see "Images" section), output collections, and image transforms (see "Workflow" section), but can also store and manage new types of assets, as defined by the user. For example binary data files obtained from a proprietary device or program, or items with no file data like a blood sample stored in a fridge. The UUID could then be used for barcode generation.

Since images are a special type of asset with extended feature support, image files may exist in various image formats, for example DICOM and NifTi[7]. Image assets are traced and recorded as to their whereabouts just like any other asset, but in addition they can be viewed, rated, converted to other image formats, and processed using image processing algorithms.

The program md5sum is used extensively throughout theHiveDB. Md5sum is designed to verify data integrity using the MD5 (Message-Digest algorithm 5) 128-bit cryptographic hash. MD5 hashes can confirm both file integrity and authenticity. Md5sum information is registered for all assets managed by the database to allow for data verification upon transfer or backup creation.

[7] http://nifti.nimh.nih.gov/



A

Project	Individual	Gender	Marker 1	Marker 2
adni	234	Male	0	2
adni	34	Female	1	17
adni	221	Female	1	38
adni	113	Male	0	89

B

Project	Individal	Timepoint	TestScore1	TestScore 2
adni	234	M00	44	11
adni	34	M12	24	17
adni	221	M00	42	22
adni	113	M06	38	20

C

Asset	Total_Volume	Region_1_Volume	Region_2_Volume
a1c168f7-5181-42e3-9ac4-921e9848f011	223000	20000	18000
0e71a8e1-432f-4560-914e-1f35134e2e40	230000	21000	18200
6382d17a-75eb-487a-95c1-9ca7cc4a0ca4	199000	18000	13000
28ff7096-5921-4960-9616-ba7b0a8ed100	200000	18080	18900

FIGURE 3 | TheHiveDB supports convention based import of tabular data. Scalar data can be imported on three levels: describing individuals **(A)** (e.g., gender, genetic data), individuals at timepoints **(B)** (e.g., clinical tests), or assets **(C)**. Since the asset belongs to a project, individual, and timepoint (e.g., activity output) the assignment can be performed automatically by just providing the unique asset ID.

In summary, assets are either created by directly uploading files via the web interface (see **Figure 2**) or by invoking activities on other assets already in the system. Assets specific to imaging projects extend the feature set of regular assets and the system provides built-in activities to derive, manage, and transform them effectively.

ARCHIVING AND AUTOMATION

DICOM management, storage, and compression

The system supports DICOM data management by means of special assets called DICOM archives. Uploaded DICOM data is packaged and compressed after relevant DICOM header information is automatically extracted. The compression ratio (uncompressed/compressed) for the lossless compression method used is around three, resulting in space savings of about 70%. Lossless compression techniques ensure that the original data can be exactly reconstructed from the compressed data. The resulting DICOM archive assets are single files containing some metadata and the entire collection of DICOM files. Once created, DICOM archives are considered immutable. Image series can be extracted as needed without any modification to the archive. Due to the deterministic nature of the unique identifiers used, they can also be migrated and imported into other HiveDB instances.

During the archival process information about all individual DICOM series is extracted and later used for automatic validation of scanning protocols. Each individual file contained in the archive is documented as a member of a DICOM archive and DICOM series including its md5sum. The information stored in the database is a reflection of the actual data found in the DICOM headers.

Metadata is stored in the database using three data domains:

1. DICOM archives – documenting the actual archive as packaged on disk.
2. DICOM series – documenting specific parameters of individual series contained in the archive.
3. DICOM files – documenting every single DICOM slice as members of the above series and archive.

Advantages of this approach include:

- Single archives instead of thousands of files on disk per study (scanning) session, resulting in significantly improved transfer speeds and file system performance.
- Significant space savings (up to 70%).
- Convenient for long time cloud storage in Amazon Glacier or offline tape storage for backup purposes (http://aws.amazon.com/glacier/).
- Content querying and information about study available through the database instead of interaction with data on disk.
- It is possible to target individual series for extraction or conversion to various image formats.
- Data verification and validation can be performed at various levels as md5sums are stored for every single DICOM file and the entire archive.
- Regardless of original scanner export convention, files can be re-organized and fed to processing pipelines in an automated fashion (The system knows which individual files make up a series, which one is the first DICOM file, etc.).
- Data provenance. The system also extracts and manages information about the scanning device used to acquire images (e.g., Manufacturer, software version, field strength, etc.). Scanners

are managed using their serial numbers and software versions, such that users can search for data acquired on specific devices.

Raw (DICOM) data de-identification/anonymization

Modern imaging systems conforming to DICOM specifications sometimes include protected health information (PHI) in the exported data. Privacy laws such as the European Commission's Directive on Data Protection and the U.S. Health Insurance Portability and Accountability Act (HIPAA) restrict the sharing of data containing PHI. These laws protect citizens but complicate the day-to-day operations of scientific collaboration. Prior to any analysis of research group data or collaboration with other groups imaging data needs to be anonymized. Without a stringent workflow, users might forget to de-identify data, or incompletely de-identify data, before using it for an analysis or even sharing it. TheHiveDB enables projects to follow privacy laws affecting medical research projects. DICOM header information of newly inserted data will be visible only to the uploading user and must be confirmed as anonymized, before data can be assigned to a project. TheHiveDB is designed to coexist with PACS systems and is by no means a replacement for a PACS system. In the workflow describing a typical imaging study theHiveDB situates itself right after either an imaging system such as a MRI system or a PACS system (unless data is available publicly or via collaborators). Network architecture and local data retrieval regulations govern the interaction of theHiveDB with PACS systems. For instance, newly acquired data still located on a PACS system can either be exported and directly uploaded via theHiveDB's upload interface or pushed to a workstation, which is registered as a HiveDB resource. PACS systems are governed by local (hospital) laws and governmental regulations. In hospital environments they may also store data for all scanned individuals, even those not to be retrieved for imaging research projects. TheHiveDB is designed to be used only with de-identified data and can easily be integrated into existing environments to enhance patient confidentiality by means of imposing a stringent workflow and data flow.

Images

Once a DICOM archive is assigned to a project, individual and timepoint, data becomes available and will be visible to all approved members of that particular project. Acquisition protocol details (e.g., echo time, repetition time, or slice thickness) for each project can be defined through the web interface, such that matching acquisitions can be extracted automatically as MR image assets. The system performs automated control of compliance with acquisition protocol details defined for any given project, as by default it rejects the extraction of acquisitions using invalid scanning parameters.

The image asset is an abstract entity representing the series of a certain scan type (i.e., T1 or T2 weighted MRI, etc.) obtained in the scanning session. The image asset inherits project specific properties during extraction from the original DICOM archive. As discussed in the "Storage Architecture" section it may be represented by actual files stored on disk (i.e., image files), at possibly various locations and a number of file formats. Currently a DICOM series may be extracted from the archive and stored as zipped DICOM data, compressed NifTi and minc[8]. Project settings determine which formats will be generated during the extraction. Image format conversion is part of the core system. Freely available converters will be added to produce additional image formats. TheHiveDB considers DICOM as the source for all conversions for native files, but will support conversion between formats if converters are available.

DATA ACCESS, PERMISSIONS, AND OWNERSHIP

Users of the theHiveDB gain access to project data by means of project memberships. Projects are collections of imaging and associated data acquired or assembled with intent to answer scientific questions. Project data is stored on resources (i.e., project servers) assigned to them. Projects have administrators authorized to grant membership to new users. Upon login users may activate any number and combination of projects they are members of, to view, add, and query data or perform quality control on both original images as well as processed output (see "Annotation" section) or request activities for assets (i.e., initiate processing on imaging data). Predefined user roles determine which actions users may perform for projects (e.g., view, create, and delete assets). A user may be allowed to only view and query data, but cannot be barred from viewing access to individual assets or specific variable collections.

Each institution will have its own limitations as to available resources and project specific restrictions. TheHiveDB accommodates this variability by letting users define where data should be located and processed. A HiveDB instance may exist on a private local network only reachable through VPN or entirely rely on cloud offerings. Constraints are defined by institutional regulations, data usage, and ownership restrictions possibly on a per project basis. Groups may even choose to have two instances of theHiveDB to separate internal and collaboration databases. **Figure 4**

[8] http://www.nitrc.org/projects/minc/

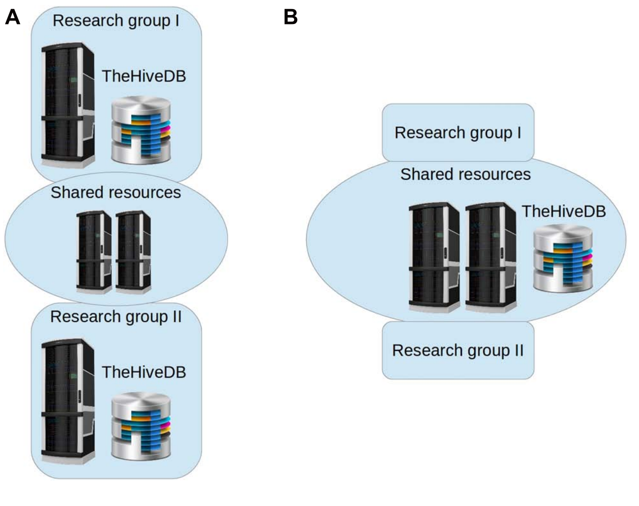


FIGURE 4 | TheHiveDB topology can be adjusted to individual requirements. Illustration of separate database instances per research lab and some compute resource sharing [**(A)** to the left] versus collaborative setting with unified architecture [**(B)** to the right].

shows topology examples for different requirements. Frequently research labs will prefer to have their own HiveDB instance with the project source data stored on an in house file system (**Figure 4A**). Collaboration in these cases will be based on sharing resources for processing purposes and possibly sharing access to subsets of projects across labs. Note, that theHiveDB database server, file servers for data store, and processing resources may well all be at different physical locations and on different servers. If however, resources are to be pooled, a unified topology arrangement (**Figure 4B**) is also a viable option.

NEUROIMAGING DATA PROCESSING ALGORITHMS, PACKAGES, AND LIBRARIES

TheHiveDB incorporates a growing number of mechanisms for data management, archival, extraction of images, and image transforms. Additionally freely available activities and pipelines are being integrated. A range of powerful neuroimaging pipelines exist today such as Freesurfer[9] and FSL[10]. Freesurfer will be used as an example application here. Briefly, the Freesurfer pipeline can be used for volumetric segmentation, cortical surface reconstruction, and cortical parcellation (Fischl et al., 2002, 2004). The procedure automatically assigns a neuroanatomical label to each voxel in an MRI volume based on probabilistic information automatically estimated from a manually labeled training set. This segmentation approach has been used for multivariate classification of AD and healthy controls (Westman et al., 2011a,c), neuropsychological-image analysis (Liu et al., 2010c, 2011), imaging-genetic analysis (Liu et al., 2010a,b), and biomarker discovery (Thambisetty et al., 2010, 2011).

TheHiveDB also provides convenient mechanisms for proprietary (or not publically accessible) processing algorithms to be integrated with their respective authors. Access is granted by these authors within the context of collaborative efforts. Any compute resource capable of ssh-2 connections can be registered in theHiveDB and used to perform tasks for specific projects (see "Data Processing and Workflow" section).

Activities may be triggered by project settings or via the application web interface. The transfer of required input files to available resources is performed automatically using the ssh-2 protocol for secure connections. Any activity requested by the system is logged and visible through the web interface job management module which provides live job queue monitoring, accounting and statistics. Upon job completion automated retrieval of processing output (e.g., output images and summary measures such as volumes and thicknesses) is also triggered by the job module.

TheHiveDB supports a number of common image management and processing activities directly. It supports external activities indirectly by automatically transferring required input files onto suitable resources and generating unique output collection identifiers for expected results. Upon completion of external processing these identifiers may be used to upload the results following naming conventions. For instance, a user experimenting with a new algorithm combining information from two types of MR images could register that activity and a resource for required input files. The system will then compute a unique identifier for each requested task and create a directory structure using these identifiers and place the required input files at the remote location within the respective directories. Jobs will be marked as completed once the user uploads properly named output files (i.e., using the identifiers computed for the task). With this method virtually any activity (including those requiring manual interaction) can be performed on existing assets (e.g., images) while output and results remain fully traceable.

For activities not yet to be registered and experimental purposes, assets can be pushed to any location registered by the requesting user for convenient examination or processing.

VISUALIZATION/ANNOTATION

In the quality control interface the user can rate both raw images and processed output. The system uses a multi rater approach, recording ratings of all authorized users separately. For quick inspection theHiveDB will create quality control images for every image or image transform visible through the web interface image library (**Figure 5**). However, images and image transforms are accessible directly in various formats or may be transferred to another resource for in-depth inspection and quality control. For instance a user may push nifti format native images for an entire project to a workstation instead of downloading them one by one in order to perform quality control. This approach allows the raters to use their preferred tools and image formats for quality evaluation. For DTI or BOLD data for instance external software is essential to perform quality control. Those images can be evaluated on a dedicated quality control station (e.g., using DTI Prep) and the results uploaded as a spreadsheet containing the image UUID as an identifier.

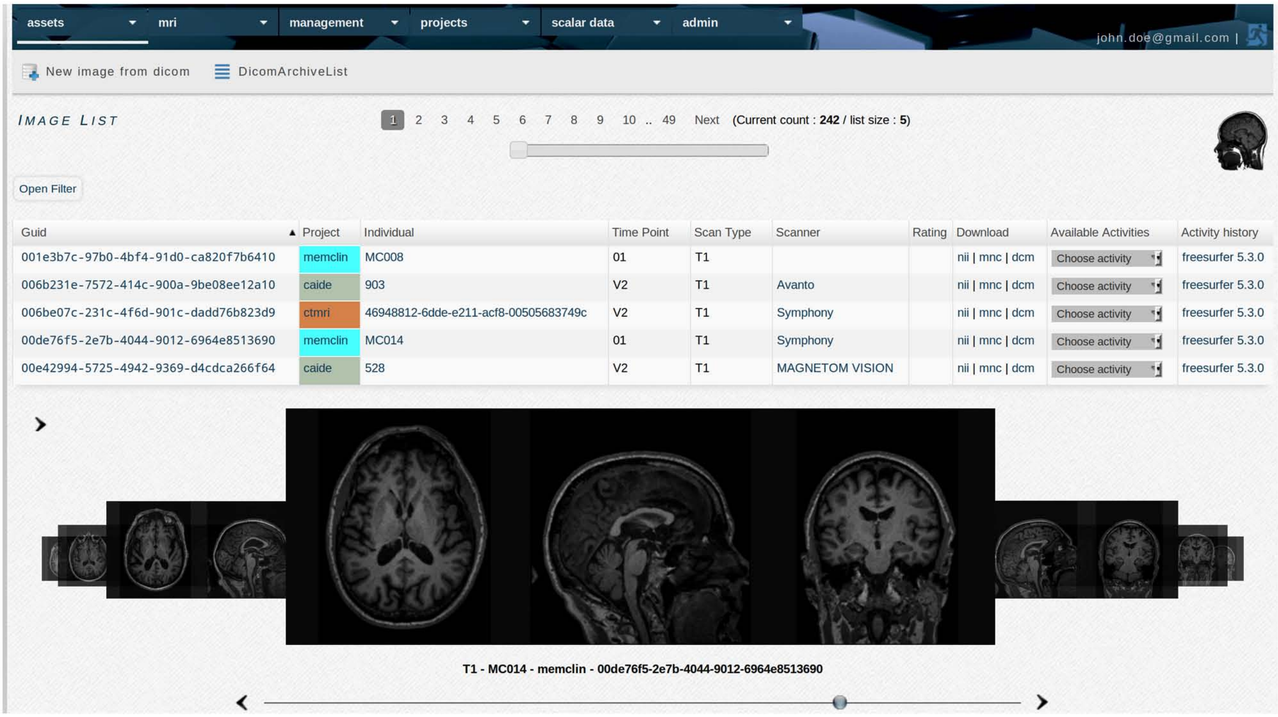
Quality control information can later be retrieved when querying the database. For example image processing for multiple images of varying quality can be compared to assess the impact of image artifacts and overall quality on processing output. Criteria for image QC for structural MRI image analysis pipelines have previously been published (Simmons et al., 2009, 2011).

DATA PROCESSING AND WORKFLOW

Computing resources (i.e., physical or virtual machines) can be managed through theHiveDB activity system. A compute resource is registered by means of providing a host (i.e., IP address, hostname) and ssh login credentials (i.e., username and password). The user registering the compute resource will be considered its owner by theHiveDB. For theHiveDB to actually utilize the resource, a resource purpose (e.g., processing resource, project server or dropbox) needs to be assigned. Choosing "processing resource" will prompt for input and output paths to be registered. At this point an "activity instance" can be created. Simplified an "activity instance" corresponds to the invocation of a specific command/program on that resource (multiple instances can be created with different parameters and environment settings passed to the command). By means of granting access to projects the resource owner manages which project data can be transferred to the resource and processed as defined in the "activity instance." TheHiveDB instance will act on behalf of the user login registered by the resource owner. To optimize resource use through

[9] http://surfer.nmr.mgh.harvard.edu/

[10] http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/



(Current count : **242** / list size : **5**)

Guid	Project	Individual	Time Point	Scan Type	Scanner	Rating	Download	Available Activities	Activity history
001e3b7c-97b0-4bf4-91d0-ca820f7b6410	memclin	MC008	01	T1			nii | mnc | dcm	Choose activity	freesurfer 5.3.0
006b231e-7572-414c-900a-9be08ee12a10	caide	903	V2	T1	Avanto		nii | mnc | dcm	Choose activity	freesurfer 5.3.0
006be07c-231c-4f6d-901c-dadd76b823d9	ctmri	46948812-6dde-e211-acf8-00505683749c	V2	T1	Symphony		nii | mnc | dcm	Choose activity	freesurfer 5.3.0
00de76f5-2e7b-4044-9012-6964e8513690	memclin	MC014	01	T1	Symphony		nii | mnc | dcm	Choose activity	freesurfer 5.3.0
00e42994-5725-4942-9369-d4cdca266f64	caide	528	V2	T1	MAGNETOM VISION		nii | mnc | dcm	Choose activity	freesurfer 5.3.0

T1 - MC014 - memclin - 00de76f5-2e7b-4044-9012-6964e8513690

FIGURE 5 | TheHiveDB image library provides access to the quality control interface and allows the user to request processing of images. List items are links to individuals, actual image data in various formats for direct download, scanner information, and activity history, etc.

collaboration without compromising processing speed, requests for external projects (i.e., from other HiveDB instances) can be assigned to a separate grid engine queue.

Within this collaborative ecosystem algorithm developers can create versioned virtual machines capable of running their tools using the cloud (e.g., Amazon cloud ec2)[11]. TheHiveDB users can run instances of these virtual machines and assign them to projects in order to take advantage of these algorithms. To illustrate the potential of this approach, consider the example of a virtual machine created in the cloud using a standard Linux installation with grid engine enabled and the Freesurfer 5.3 package added. At this point the compute resource can be registered by its owner in any HiveDB instance. Furthermore the resource owner can register activity instances and assign them to projects in order to authorize them to use the resource. An activity instance defines the algorithm or activity to be used (e.g., Freesurfer version 5.3), the activity parameters to be used and the projects authorized to request it. Activity parameters are command line arguments to be passed to the command to be executed (e.g., Freesurfer can run with "-all-mprage-nuintensitycor-3 T" for a project with 3 T imaging data and with a basic exploratory argument like "-recon1" for a second project).

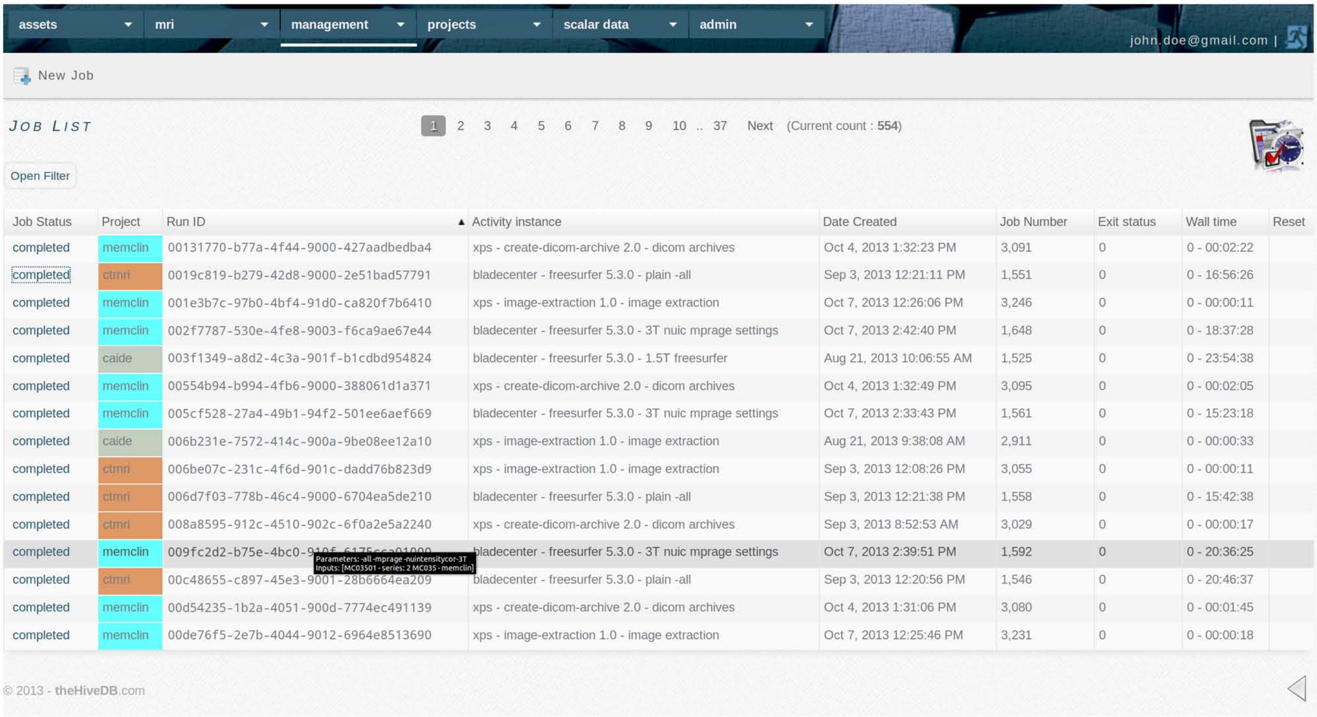
While the above can also be achieved with a conventional physical server, the cloud approach has a number of advantages. Apart from minimal storage costs a cloud image only incurs cost to the user when it is used. It doesn't physically break down and its hardware can improve over time as newer technology is made available by the cloud provider. When newer versions of algorithms are released images of previous versions may be kept for ongoing projects still requiring them. This is especially relevant when different versions of software packages cannot be installed on the same physical system.

PROVENANCE AND META-DATA MANAGEMENT

Imaging source data is fully documented as described in the "DICOM management, storage, and compression" section. All assets produced or derived within the database system are traceable using the job sub-system (**Figure 6**). Every activity within theHiveDB consumes input and produces an output collection (i.e., a compressed archive file containing the individual results obtained from an image processing activity). The output of any activity is considered to be a collection containing at least one item. If members of an output collection have been defined they can be extracted automatically by the system (e.g., tissue classification result image obtained from a processing pipeline).

This concept provides full traceability of newly generated data, transparency for all project members, and an ever current inventory to assess project progress. A fully-fledged activity system is a prerequisite for enabling advanced processing for extremely large amounts of imaging data. Algorithm comparison with regard to stability across versions and vulnerability to image artifacts can be performed. For instance, all images obtained from an individual during a MRI session can be processed individually or

[11] http://aws.amazon.com/ec2/



JOB LIST

1 2 3 4 5 6 7 8 9 10 .. 37 Next (Current count : **554**)

Job Status	Project	Run ID ▲	Activity instance	Date Created	Job Number	Exit status	Wall time	Reset
completed	memclin	00131770-b77a-4f44-9000-427aadbedba4	xps - create-dicom-archive 2.0 - dicom archives	Oct 4, 2013 1:32:23 PM	3,091	0	0 - 00:02:22	
completed	ctmri	0019c819-b279-42d8-9000-2e51bad57791	bladecenter - freesurfer 5.3.0 - plain -all	Sep 3, 2013 12:21:11 PM	1,551	0	0 - 16:56:26	
completed	memclin	001e3b7c-97b0-4bf4-91d0-ca820f7b6410	xps - image-extraction 1.0 - image extraction	Oct 7, 2013 12:26:06 PM	3,246	0	0 - 00:00:11	
completed	memclin	002f7787-530e-4fe8-9003-f6ca9ae67e44	bladecenter - freesurfer 5.3.0 - 3T nuic mprage settings	Oct 7, 2013 2:42:40 PM	1,648	0	0 - 18:37:28	
completed	caide	003f1349-a8d2-4c3a-901f-b1cdbd954824	bladecenter - freesurfer 5.3.0 - 1.5T freesurfer	Aug 21, 2013 10:06:55 AM	1,525	0	0 - 23:54:38	
completed	memclin	00554b94-b994-4fb6-9000-388061d1a371	xps - create-dicom-archive 2.0 - dicom archives	Oct 4, 2013 1:32:49 PM	3,095	0	0 - 00:02:05	
completed	memclin	005cf528-27a4-49b1-94f2-501ee6aef669	bladecenter - freesurfer 5.3.0 - 3T nuic mprage settings	Oct 7, 2013 2:33:43 PM	1,561	0	0 - 15:23:18	
completed	caide	006b231e-7572-414c-900a-9be08ee12a10	xps - image-extraction 1.0 - image extraction	Aug 21, 2013 9:38:08 AM	2,911	0	0 - 00:00:33	
completed	ctmri	006be07c-231c-4f6d-901c-dadd76b823d9	xps - image-extraction 1.0 - image extraction	Sep 3, 2013 12:08:26 PM	3,055	0	0 - 00:00:11	
completed	ctmri	006d7f03-778b-46c4-9000-6704ea5de210	bladecenter - freesurfer 5.3.0 - plain -all	Sep 3, 2013 12:21:38 PM	1,558	0	0 - 15:42:38	
completed	ctmri	008a8595-912c-4510-902c-6f0a2e5a2240	xps - create-dicom-archive 2.0 - dicom archives	Sep 3, 2013 8:52:53 AM	3,029	0	0 - 00:00:17	
completed	memclin	009fc2d2-b75e-4bc0-910f-6175cca01000	bladecenter - freesurfer 5.3.0 - 3T nuic mprage settings	Oct 7, 2013 2:39:51 PM	1,592	0	0 - 20:36:25	
completed	ctmri	00c48655-c897-45e3-9001-28b6664ea209	bladecenter - freesurfer 5.3.0 - plain -all	Sep 3, 2013 12:20:56 PM	1,546	0	0 - 20:46:37	
completed	memclin	00d54235-1b2a-4051-900d-7774ec491139	xps - create-dicom-archive 2.0 - dicom archives	Oct 4, 2013 1:31:06 PM	3,080	0	0 - 00:01:45	
completed	memclin	00de76f5-2e7b-4044-9012-6964e8513690	xps - image-extraction 1.0 - image extraction	Oct 7, 2013 12:25:46 PM	3,231	0	0 - 00:00:18	

Parameters: -all -mprage -nuintensitycor-3T
Inputs: [MC03501 - series: 2 MC035 - memclin]

© 2013 - **theHiveDB**.com

FIGURE 6 | The activity system keeps a track record of all activities performed by the system or requested by users. It communicates with remote resources for status updates and retrieves output collections automatically. Accounting information is compiled using the grid scheduler's accounting output.

in combination. Additionally multiple versions of the processing algorithm can be used and results compared.

USE CASE

To illustrate the application of theHiveDB consider the scenario where researchers wish to download and subsequently analyze raw data from the ADNI study, a large North American study which includes 1.5 T MRI, 3 T MRI, FDG, and amyloid PET, together with CSF samples, clinical, cognitive, neuropsychological, and genetic data.

The raw imaging data for the study can be downloaded from the LONI distribution system[12] in the form of collections of raw DICOM data with xml files for each image series. Additionally data from other modalities such as cognitive tests, demographic information, genetic data, and CSF data can be downloaded in tabular form.

The user has the following requirements for image databasing and analysis:

- Users need to be able to perform operations like tabular data import and imaging data upload through the web interface.
- No alterations of the database structure (i.e., adding tables) should be needed for newly added variables or results derived using imaging data processing algorithms.
- Tabular data import should be possible instead of data re-entry through a web interface.
- Users need to be able to deactivate projects. Regardless of a user's authorization to see data (e.g., has access to 40 projects) the user needs to be able to activate only those of current interest.
- Data needs to be accessible directly by project members via the database web interface without the need for an intermediary (e.g., a database manager retrieving data) when image processing is desired.
- The system needs to support multiple image file formats as the inputs and outputs of different image analysis pipelines and manage data effectively as opposed to merely registering file pointers for a single file format.
- Processing of as many images as available for any given timepoint using algorithms and pipelines available is required, for example processing the two T1 volume images acquired as part of ADNI-1 and ADNI-2.
- Integration of existing infrastructure and processing capabilities with automated processing as triggered by the database system.
- As new versions of image analysis pipelines become available it must be possible to maintain multiple versions of both the algorithms and the results of the analysis pipelines (for example Freesurfer versions 5.1 and 5.3).

TheHiveDB was designed to provide the feature set of similar distribution and collection systems in the neuroimaging domain, but extending them to a more complete framework with the above requirements in mind.

[12] https://ida.loni.usc.edu/login.jsp

SCALAR DATA IMPORT

TheHiveDB allows for collation of existing data by simply uploading spreadsheets with scalar data. Variables are grouped into variable collections. The import is governed by conventions. Using existing collection names will add data to collections using the first line as field names. If variables are identified as members of the same collection they are queryable across projects (e.g., if Mini-mental state examination MMSE data is always imported using the same field names).

Variables can be imported (see **Figure 3**) and may later be queried at these three levels:

1. Describing individuals (permanently) like some genetic data or gender.
2. Describing individuals at specific timepoints (e.g., clinical or cognitive tests).
3. Describing assets obtained to assess individuals at specific timepoints (e.g., MR images or volume results from processing pipelines).

Via theHiveDB web interface a user creates a new project "adni" and creates or assigns an existing compute resource for project data (i.e., project server). Disk space of the project server will be used as primary location for all project data assigned to this project. The user registers timepoints (i.e., adni visit identifiers) and defaults for desired image format conversion. For this example DICOM and nifti are chosen as available formats.

The user downloads a list of ADNI study participants and creates a spreadsheet containing the following fields: project, SiteId. Gender, and DateOfBirth. Gender and DateOfBirth are not mandatory, but may be provided. The file is renamed to "adni.individuals.list.csv" and uploaded. All individuals are now registered and assigned to the "adni" project. Following the examples outlined in **Figure 3** more data describing the individual permanently or describing the individual at a specific timepoint may be uploaded.

IMAGING DATA UPLOAD PREPARATION

Raw ADNI imaging data is downloaded via the LONI distribution system, resulting in a folder structure based on the ADNI series identifier. Auxiliary xml files with summary information about the individual, series and assignment to a visit identifier will be found at the top level of the folder structure.

For smaller projects data would be uploaded directly to theHiveDB web interface marked as anonymized and assigned to projects, individuals, and timepoints. In view of the amount of data downloaded [ADNI MP-RAGE (T1) data occupies 400 GB of disk space] and since data is known to be anonymized, an alternative route to DICOM archive creation is used. Based on information from the ADNI xml files a spreadsheet containing the following columns is created:

- Project (i.e., "adni")
- Individual (i.e., the PatientID as found in DICOM header or xml file)
- TimePoint (i.e., the visit identifier found in the xml file)
- SourceLocation (i.e., the location data has been downloaded to)
- TargetLocation (i.e., the location where the DicomArchive and descriptor file is to be created. If the project server location is available the user may choose it to avoid data transfers.)

TheHiveDB provides a convenience function for large data collections. Upon upload the spreadsheet (in this case ~16,000 rows) will be converted into a job script, which can be submitted to the queue for HiveDB DICOM archive creation. This activity requires no connection to the HiveDB instance and can run directly on the Linux machine already hosting the downloaded data. UUIDs are computed using the same mechanism as within theHiveDB and for every folder containing DICOM data a compressed DICOM archive and a supplementary descriptor file in JSON format[13] is created. For the above example the procedure takes on average 5 s per folder. Within 3 h on a desktop machine (eight cores) this process transforms almost three million single DICOM slices into about 16,000 completely documented and compressed DICOM archives.

DATA IMPORT AND ORGANIZATION

The descriptor files are subsequently uploaded via theHiveDB web interface resulting in DICOM archives being automatically created and assigned to project, individuals, and timepoints. The user defines at least one scanning protocol for the "adni" project, such that the system can automatically identify T1 data. After a test search the user confirms the protocol as valid leading to automatic extraction of all MR image assets and the creation of downloadable image files in DICOM and nifti format. All data is now available to project members through the web interface.

DATA PROCESSING

The user registers a processing resource (i.e., an existing cluster the user has access to) and defines two activity instances through the web interface (One instance with parameters to be used for 1.5 T data labeled "Freesurfer-5.1 1.5 T" and another one for 3.0 T data labeled "Freesurfer-5.1 3.0 T").

The user now selects the current session preferences panel and deactivates all projects other than "adni." In the image library the user now searches for T1, 1.5 T, and timepoint M00 (i.e., baseline) data and chooses "Freesurfer-5.1 1.5 T" as activity to apply to all elements found, followed by the same procedure for 3.0 T data. Standard processing time for Freesurfer is in the vicinity of 16 h per image. Processing all 16,000 images on the 100 core cluster currently providing processing for the production database will take about 3 months. For this reason timepoints will be submitted in sequence in order to start analysis on data as it becomes available. The user may now log off.

TheHiveDB will create job files (using computed UUIDs for names), transfer inputs to the processing cluster and submit jobs. It will monitor the queue and upon job completion retrieve an output collection (a tar file) containing the results for every single job.

If new versions of the pipeline (e.g., Freesurfer 5.3) become available, the creation of additional activity instances is required. The steps above are repeated with the new version of the pipeline.

[13] http://www.json.org/

Since the activity version is part of the computation of UUIDs, new unique identifiers for outputs will be provided by the system.

The Freesurfer pipeline outputs a multitude of different measures (Fjell et al., 2009; Walhovd et al., 2011; Westman et al., 2013), which need to be queried and combined for analysis with data from other modalities. Since Freesurfer is directly supported by the database, volume extraction will be performed automatically and all volumes will be registered in a variable collection labeled Freesurfer-5.1. The user may now query those volumes in conjunction with other data uploaded via tabular data import. If the user produces additional measures using external methods to compute scalar values those may be uploaded following conventions depicted in **Figure 3**. TheHiveDB aggregates data from these different modalities automatically and combines it with image processing results, such that research problems can be addressed without the need to manually manage and merge spreadsheets.

DISCUSSION

RELATED WORK

TheHiveDB has been developed to advance imaging efforts in a context where more and more data is available to researchers either by means of in house acquisition or more frequently by means of collaboration. The latter includes the growing number of publicly available collections of imaging data such as the ADNI; Jack et al., 2008; Weiner et al., 2010) and AddNeuroMed (Lovestone et al., 2009, 2007).

Most of these collections use a distribution system (such as the LONI ADNI archive)[14] to disseminate data. In these systems assets (the raw imaging data) and associated data from other modalities are readily accessible and frequently processed data (output collections) can also be downloaded. The LONI image data archive provides scalar data organized into spreadsheets. An accompanying data dictionary helps clarify the meaning of variable names contained in these spreadsheets. For any given group of variables (typically a questionnaire or the scalar results of some processing or other analysis of data) spreadsheets can be downloaded. It is up to the researcher to match data from different cohorts (Westman et al., 2011b) or modalities (Westman et al., 2012) prior to any data analysis being undertaken using the raw data and images. Ever changing spreadsheets have to be organized, merged, and maintained. The creation of subsets of data to investigate specific research questions remains a cumbersome process.

Other systems like the LORIS system (Das et al., 2012) focus on scalar data collection for relatively homogeneous ongoing studies. The LORIS system needs to be customized at the database structural level, before its web interface can be used as a data entry system by participating sites. Each addition of tabular data implies a change to the database structure to store data for newly added variables. While the LORIS query interface is able to match some MRI data to clinical variables, the imaging component remains an afterthought due to the system's architectural conception as scalar data entry system. The LORIS web interface does not allow distribution of data directly as file data is only referenced in the database and solely accessible via command line interfaces on servers hosting the actual data.

For handling ongoing data collection and data entry the REDCap (Obeid et al., 2013) system appears to be a more feature complete and convenient system. REDCap is designed to comply with HIPAA regulations and can be quickly adjusted to cover all aspects of research data capture.

The LONI pipeline (Dinov et al., 2010) provides a collection of neuroimaging tools for computational scientists. It allows for workflow creation and execution via Pipeline Web Start (PWS)[15].

The XNAT (Marcus et al., 2007) system and its Python client library PyXNAT (Schwartz et al., 2012) represent the web services approach to neuroimaging databases. Neuroimaging data is modeled through XML schemas and a representational state transfer Application Program Interface (REST API) allows software developers to programmatically interact with the database system.

Research labs can struggle with how to organize the ever growing collections of data. Most neuroimaging databases consequently provide a container based approach with a more or less predefined structure to organize data. This approach works well to organize data as long as the data stays within the realm of control of the database system. A user who downloads a set of image files to perform processing temporarily breaks the way data is structured in the database. If files have no unique identifiers or can be identified by means of header tags or md5sums the interaction of the researcher with the database system is rapidly disturbed. Data needs to be reorganized and the database needs to be updated with newly created results. Unfortunately this implies frequent changes to the actual database structure and/or creation of XML schemata.

Most of the above mentioned database systems are designed as containers or data inventories. The container approach works well for data entry systems where the size of a prospective study warrants the effort of customization, but they are frequently limited to tabular data collection. The other approaches require the user to interact programmatically with the database system to retrieve data and repopulate with results.

TheHiveDB goes beyond these approaches. It offers ways to organize data beyond simple storage. Imaging data assets are enhanced with features to simplify the researcher's interaction with the data (See sections "DICOM management, storage, and compression" and "Images"). While programmatically interacting with theHiveDB is an option for advanced users (theHiveDB is a RESTful resource) the framework aims to accompany and support the researcher in daily activities and explorations. Standard activities can be automatically performed for any new project with existing resources and new activities can be explored with the help of the system. All assets remain identifiable within and outside the system. Even for external activities the identifier creation keeps expected results traceable. This way manual steps or external resources for free image processing can be integrated.

TheHiveDB implements the main ideas of other activity and workflow systems. Tools and algorithms are available to the researcher and can be applied to available data. To warrant

[14] http://adni.loni.usc.edu/data-samples/access-data/

[15] http://pipeline.loni.usc.edu/products-services/pws/

consistency without compromising progress theHiveDB requires all activities to be versioned.

The primary shortcoming of some neuroimaging frameworks is their insufficient support for file data (assets). Neuroimaging research is an active field. In order to progress imaging assets need to be available and accessible to those working with them. The unique identifiers within theHiveDB constitute tracking or serial numbers for assets. The web interface acts like a tracking system providing appropriate information. For images this may be the scanner or protocol used or assessments of image quality. For output from any activity the entire process leading to its generation is traceable. The system is not designed to force all data into a container. It encourages the interaction with the researchers by letting them experiment with assets to perform activities not (yet) supported by the system. It even provides the possibility to re-integrate results by allowing for external activities where the user needs to provide the activity output by means of uploading it, using the identifier provided by the system.

Interacting programmatically with theHiveDB API remains a possibility for the so inclined power user, but it is not a requirement for researchers. The ability to voluntarily disable access to projects throughout the system can greatly simplify the researcher's day to day interaction with the system. It can be frustrating to always have to set additional filters in order not to be exposed to all data one is authorized to see.

FUTURE DEVELOPMENT

TheHiveDB has been conceived and developed as a data aggregation system. While it currently supports scalar data import, it would be desirable for theHiveDB to interface directly with clinical data entry systems. Especially with systems allowing non-programmers to quickly create forms for tabular data collection like the REDCap application.

ThHiveDB's activity system supports activity creation based on asset types. While it is presently only used for image processing it would be conceivable to integrate workflows for other types of data (e.g., by supporting genetic data processing).

TheHiveDB allows users to directly access MR images in their preferred format to be visualized for quality control purposes. While we favor direct access to images in user definable formats and the possibility to push entire collections to dedicated quality control stations, the inclusion of a web based viewer with 3D capabilities may be desirable for some users. The integration of imageJ[16] could provide additional convenience to users in this regard. Nielsen's heuristics have influenced the design and development of theHiveDB and it has been developed in continuous interaction with future users. However, a formal evaluation of the system would be desirable.

While some architectural design elements might hint toward a federated database system currently only data exchange and migration/fusion is planned. Data ownership concerns and the protected nature of imaging database as discussed in the "Data access, permissions, and ownership" section make this a more likely scenario.

CONCLUSION

At the topological level theHiveDB provides the integration of different components – a solid database engine combined with secure data store and an activity system for data processing purposes. The application is flexible to be adapted to individual requirements and available resources without the need to customize its database tables and structure.

TheHiveDB provides extensive cross domain integration. For tabular/scalar data, convention based import (i.e., using specific column arrangements) allows for swift integration of data already available in spreadsheets or textual form.

The asset management system provides support tailored to the particular needs of brain imaging projects. But what is more, it is also capable of integrating newly defined asset types. The generation of unique identifiers extends to any type of uploaded data and provides data integrity verification and management with storage, transfer, backup, and availability. This approach clears the way for integration of imaging workflow with other types of workflow based on custom asset types.

On an architectural level theHiveDB is capable of integrating distributed systems. Each "HiveDB" has its own unique ID. Frequently individual research groups will have their own HiveDB instance (see **Figure 4A**), but share resources for activities (i.e., data processing). Additionally cloud resources can be enabled by algorithm developers to be used by those instances of theHiveDB. Project data will in most cases be stored on local resources, but long term cloud backup (e.g., Amazon glacier) for both raw imaging data and processed output is another viable option.

TheHiveDB represents another step toward creating a complete neuroimaging research framework. It provides easy access to data just like traditional distribution systems and offers the convenience of multi modal querying.

A key aim of theHiveDB is to enable collaborations. It does so by providing a framework for neuroimaging projects based on sound data management, organization, and documentation. Upon that base rests an activity system allowing for automation and resource sharing while ensuring full traceability of activities and outputs. With its asset management and activity system it establishes a powerful ecosystem for collaborative work and resource sharing in continuous interaction with the researcher.

The inclusion of standard communication protocols and job schedulers eliminates the need for a human data manager needed in most of the other systems available to date. TheHiveDB knows where project data is supposed to be stored and where it can be processed. It is capable of performing its own data transfers and request activities/processing to that effect.

The system has been designed to interact with the researcher in a (human) way that does not require the acquisition of database query language skills or programming proficiency.

AUTHOR CONTRIBUTIONS

J-Sebastian Muehlboeck, Eric Westman, and Andrew Simmons all contributed to the concept of TheHiveDB, J-Sebastian Muehlboeck is the programmer and developer of the system. J-Sebastian Muehlboeck, Eric Westman, and Andrew Simmons all contributed to the writing and editing of the manuscript.

[16] http://rsb.info.nih.gov/ij/index.html

ACKNOWLEDGMENTS

This project was supported by the Biomedical Research Centre for Mental Health at South London and Maudsley NHS Foundation Trust and Institute of Psychiatry, King's College London. Andrew Simmons was also supported by funds from the Biomedical Research Unit for Dementia and Alzheimer's Research UK. Also thanks to the Strategic Research Programme in Neuroscience at Karolinska Institutet (StratNeuro) and Swedish Brain Power.

REFERENCES

Das, S., Zijdenbos, A. P., Vins, D., Harlap, J., and Evans, A. C. (2012). LORIS: a web-based data management system for multi-center studies. *Front. Neuroinform.* 5:37. doi: 10.3389/fninf.2011.00037

Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., et al. (2010). Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline. *PLoS ONE* 5:e13070. doi: 10.1371/journal.pone.0013070

Ellis, K. A., Bush, A. I., Darby, D., De Fazio, D., Foster, J., Hudson, P., et al. (2009). The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of Alzheimer's disease. *Int. Psychogeriatr.* 21, 672–687. doi: 10.1017/S1041610209009405

Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., et al. (2002). Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron* 33, 341–355. doi: 10.1016/S0896-6273(02)00569-X

Fischl, B., Van Der Kouwe, A., Destrieux, C., Halgren, E., Segonne, F., Salat, D. H., et al. (2004). Automatically parcellating the human cerebral cortex. *Cereb. Cortex* 14, 11–22. doi: 10.1093/cercor/bhg087

Fjell, A. M., Westlye, L. T., Amlien, I., Espeseth, T., Reinvang, I., Raz, N., et al. (2009). High consistency of regional cortical thinning in aging across multiple samples. *Cereb. Cortex* 19, 2001–2012. doi: 10.1093/cercor/bhn232

Jack, C. R. Jr., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., et al. (2008). The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods. *J. Magn. Reson. Imaging* 27, 685–691. doi: 10.1002/jmri.21049

Liu, Y., Paajanen, T., Westman, E., Wahlund, L. O., Simmons, A., Tunnard, C., et al. (2010a). Effect of APOE epsilon4 allele on cortical thicknesses and volumes: the AddNeuroMed study. *J. Alzheimers. Dis.* 21, 947–966. doi: 10.3233/JAD-2010-100201

Liu, Y., Paajanen, T., Westman, E., Zhang, Y., Wahlund, L. O., Simmons, A., et al. (2010b). APOE epsilon2 allele is associated with larger regional cortical thicknesses and volumes. *Dement. Geriatr. Cogn. Disord* 30, 229–237. doi: 10.1159/000320136

Liu, Y., Paajanen, T., Zhang, Y., Westman, E., Wahlund, L.-O., Simmons, A., et al. (2010c). Analysis of regional MRI volumes and thicknesses as predictors of conversion from mild cognitive impairment to Alzheimer's disease. *Neurobiol. Aging* 31, 1375–1385. doi: 10.1016/j.neurobiolaging.2010.01.022

Liu, Y., Paajanen, T., Zhang, Y., Westman, E., Wahlund, L. O., Simmons, A., et al. (2011). Combination analysis of neuropsychological tests and structural MRI measures in differentiating AD, MCI and control groups-The AddNeuroMed study. *Neurobiol. Aging* 32, 1198–1206. doi: 10.1016/j.neurobiolaging.2009.07.008

Lovestone, S., Francis, P., Kloszewska, I., Mecocci, P., Simmons, A., Soininen, H., et al. (2009). AddNeuroMed;The European Collaboration for the discovery of novel biomarkers for Alzheimer's disease. *Ann. N. Y. Acad. Sci.* 1180, 36–46. doi: 10.1111/j.1749-6632.2009.05064.x

Lovestone, S., Francis, P., and Strandgaard, K. (2007). Biomarkers for disease modification trials – the innovative medicines initiative and AddNeuroMed. *J. Nutr. Health Aging* 11, 359–361.

Marcus, D., Olsen, T., Ramaratnam, M., and Buckner, R. (2007). The extensible neuroimaging archive toolkit. *Neuroinformatics* 5, 11–33. doi: 10.1385/NI:5:1:11

Obeid, J. S., McGraw, C. A., Minor, B. L., Conde, J. G., Pawluk, R., Lin, M., et al. (2013). Procurement of shared data instruments for Research Electronic Data Capture (REDCap). *J. Biomed. Inform.* 46, 259–265. doi: 10.1016/j.jbi.2012.10.006

Schwartz, Y., Barbot, A., Thyreau, B., Frouin, V., Varoquaux, G., Siram, A., et al. (2012). PyXNAT: XNAT in python. *Front. Neuroinform.* 6:12. doi: 10.3389/fninf.2012.00012

Simmons, A., Westman, E., Muehlboeck, S., Mecocci, P., Vellas, B., Tsolaki, M., et al. (2009). MRI measures of Alzheimer's disease and the AddNeuroMed Study. *Ann. N. Y. Acad. Sci.* 1180, 47–55. doi: 10.1111/j.1749-6632.2009.05063.x

Simmons, A., Westman, E., Muehlboeck, S., Mecocci, P., Vellas, B., Tsolaki, M., et al. (2011). The AddNeuroMed framework for multi-centre MRI assessment of longitudinal changes in Alzheimer's disease: experience from the first 24 months. *Int. J. Geriatr. Psychiatry* 26, 75–82. doi: 10.1002/gps.2491

Thambisetty, M., Simmons, A., Hye, A., Campbell, J., Westman, E., Zhang, Y., et al. (2011). Plasma biomarkers of brain atrophy in Alzheimer's disease. *PLoS ONE* 6:e28527. doi: 10.1371/journal.pone.0028527

Thambisetty, M., Simmons, A., Velayudhan, L., Hye, A., Campbell, J., Zhang, Y., et al. (2010). Association of plasma clusterin concentration with severity, pathology, and progression in Alzheimer disease. *Arch. Gen. Psychiatry* 67, 739–748. doi: 10.1001/archgenpsychiatry.2010.78

Van Horn, J. D., and Toga, A. W. (2009). Is it time to re-prioritize neuroimaging databases and digital repositories? *Neuroimage* 47, 1720–1734. doi: 10.1016/j.neuroimage.2009.03.086

Walhovd, K. B., Westlye, L. T., Amlien, I., Espeseth, T., Reinvang, I., Raz, N., et al. (2011). Consistent neuroanatomical age-related volume differences across multiple samples. *Neurobiol. Aging* 32, 916–932. doi: 10.1016/j.neurobiolaging.2009.05.013

Weiner, M. W., Aisen, P. S., Jack, C. R. Jr., Jagust, W. J., Trojanowski, J. Q., et al. (2010). The Alzheimer's disease neuroimaging initiative: progress report and future plans. *Alzheimers Dement* 6, 202.e7–211.e7. doi: 10.1016/j.jalz.2010.03.007

Westman, E., Aguilar, C., Muehlboeck, J. S., and Simmons, A. (2013). Regional magnetic resonance imaging measures for multivariate analysis in Alzheimer's disease and mild cognitive impairment. *Brain Topogr.* 26, 9–23. doi: 10.1007/s10548-012-0246-x

Westman, E., Cavallin, L., Muehlboeck, J. S., Zhang, Y., Mecocci, P., Vellas, B., et al. (2011a). Sensitivity and specificity of medial temporal lobe visual ratings and multivariate regional MRI classification in Alzheimer's disease. *PLoS ONE* 6:e22506. doi: 10.1371/journal.pone.0022506

Westman, E., Simmons, A., Muehlboeck, J. S., Mecocci, P., Vellas, B., Tsolaki, M., et al. (2011b). AddNeuroMed and ADNI: similar patterns of Alzheimer's atrophy and automated MRI classification accuracy in Europe and North America. *Neuroimage* 58, 818–828. doi: 10.1016/j.neuroimage.2011.06.065

Westman, E., Wahlund, L.-O., Foy, C., Poppe, M., Cooper, A., Murphy, D., et al. (2011c). Magnetic resonance imaging and magnetic resonance spectroscopy for detection of early Alzheimer's disease. *J. Alzheimers Dis.* 26, 307–319. doi: 10.3233/JAD-2011-0028

Westman, E., Muehlboeck, J. S., and Simmons, A. (2012). Combining MRI and CSF measures for classification of Alzheimer's disease and prediction of mild cognitive impairment conversion. *Neuroimage* 62, 229–238. doi: 10.1016/j.neuroimage.2012.04.056

Conflict of Interest Statement: J-Sebastian Muehlboeck is lead developer and president of J-S Muehlboeck Inc. Eric Westman and Andrew Simmons have no conflict of interest.

Received: 31 October 2013; accepted: 19 December 2013; published online: 06 January 2014.

Citation: Muehlboeck J-S, Westman E and Simmons A (2014) TheHiveDB image data management and analysis framework. Front. Neuroinform. 7:49. doi: 10.3389/fninf.2013.00049

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 30 April 2014
doi: 10.3389/fninf.2014.00047

Structure-centered portal for child psychiatry research

Pallavi Rane*, Christian Haselgrove, Steven M. Hodge, Jean A. Frazier and David N. Kennedy

Child and Adolescent NeuroDevelopment Initiative, Department of Psychiatry, University of Massachusetts Medical School, Worcester, MA, USA

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Dimiter Prodanov, IMEC, Belgium
Daniel Gardner, Weill Cornell Medical College, USA

***Correspondence:**
Pallavi Rane, Child and Adolescent NeuroDevelopment Initiative, Department of Psychiatry, University of Massachusetts Medical School, Suite 310, 55 N Lake Ave., Worcester, MA 01655, USA
e-mail: pallavi.rane@gmail.com

The real world needs of the clinical community require a domain-specific solution to integrate disparate information available from various web-based resources for data, materials, and tools into routine clinical and clinical research setting. We present a child-psychiatry oriented portal as an effort to deliver a knowledge environment wrapper that provides organization and integration of multiple information and data sources. Organized semantically by resource context, the portal groups information sources by context type, and permits the user to interactively "narrow" or "broaden" the scope of the information resources that are available and relevant to the specific context. The overall objective of the portal is to bring information from multiple complex resources into a simple single uniform framework and present it to the user in a single window format.

Keywords: neurodevelopmental disorders, data integration, knowledge environment, MRI, neuroinformatics

INTRODUCTION

Neuroimaging studies performed with specific hypothesis in mind are highly informative for learning details about human brain development, elucidating the etiology of numerous psychiatric disorders, and developing ways to remedy them. However, disorder-focused neuroimaging studies have a very precise and narrow objective when it comes to developing a broad understanding the human brain. The data collected during these individual studies can be a resource for extracting additional information about the disorder or the human brain in general. This general concept regarding the latent-content of research data has led to development of numerous neuroimaging data sharing resources, such as NDAR (Hall et al., 2012), NIH Pediatric Database (Evans, 2006), CANDIShare (Kennedy et al., 2012), and ADNI (Jack et al., 2008), which make neuroimaging data available to interested users. In addition to neuroimaging data, hundreds of data and information resources are also available that support dissemination of information related to literature, genetic, derived metadata results, etc. about the brain in health and disease. Despite the burgeoning set of resources hosting research information, attempts to query across these distributed resources are daunting due to variation in the underlying data models, schema and interfaces.

While methods to improve the accessibility of these disparate data resources are underway, an additional consideration needs to be paid to the end user. The Neuroscience Information Framework (NIF) portal (Gardner et al., 2008; Cachat et al., 2012) is an effort to integrate web-based neuroscience resources such as data, materials, and tools. In addition to this general and comprehensive infrastructure, domain-specific solutions are needed in order to meet the real-world needs of the various clinical communities where there is a need to incorporate and integrate these disparate information resources into the routine clinical and clinical research setting.

In this paper we describe the design of a child-psychiatry oriented portal as an effort to deliver a knowledge environment wrapper that provides organization and integration of multiple information sources. Organized semantically by resource context, the portal groups information sources by context type, and permits the user to interactively "narrow" or "broaden" the scope of the information resources that are available and relevant to the specific context. The overall objective of the portal is to bring information from multiple complex resources into a simple single uniform framework and present it to the user in a single interface from which they can easily continue to explore the relevant resources as needed. We will review the conceptual design, describe the methods of implementation, and provide examples of its operation. This will be followed by a discussion of the impact, impediments and future prospects for this type of approach.

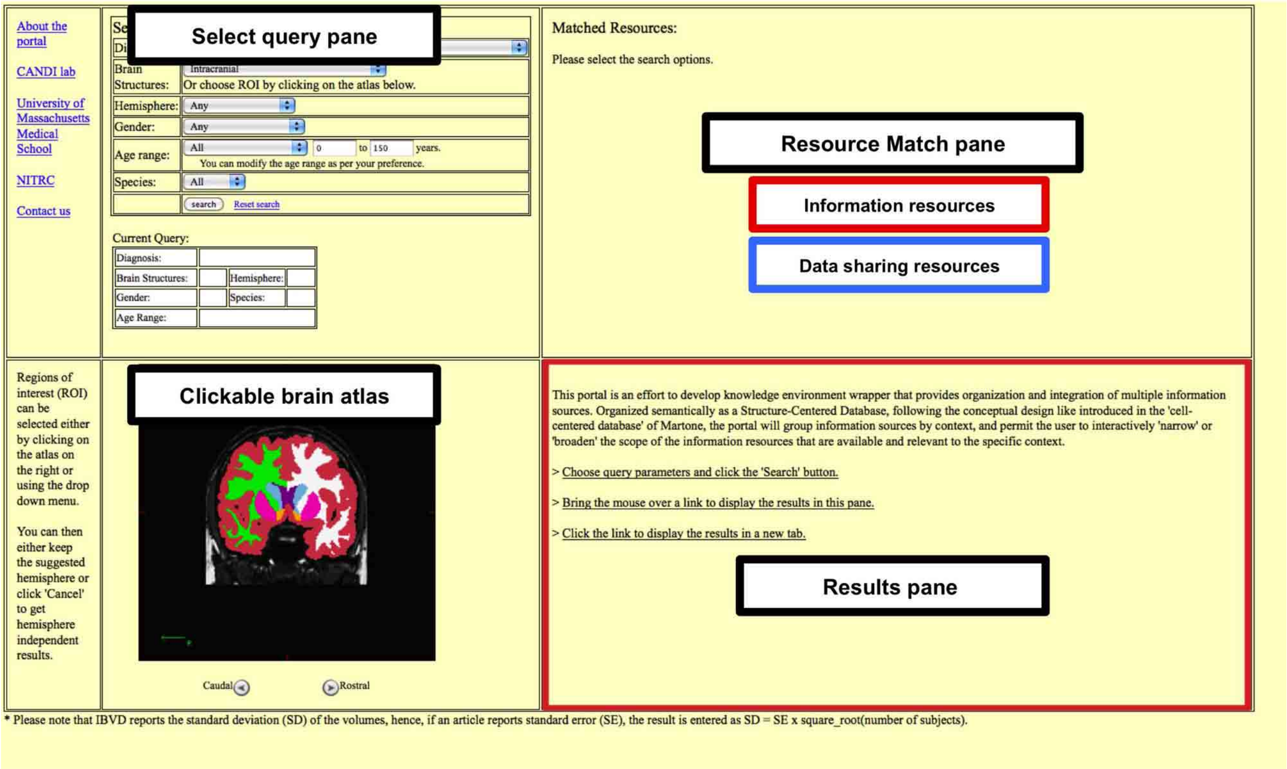
METHODS

In this section we review the conceptual design, followed by the practical implementation of the portal. We emphasize the extensible nature of the design, and highlight how content from existing resources is accessed under a common user framework.

PORTAL DESCRIPTION

The overall system is designed such that a user can generate specific classes of query, identify the various resources that can provide information relevant to the query and then view the results from each of the resources.

The portal front end has a four-pane window format (**Figure 1**): Select Query, Anatomic Atlas, Resource Match and Results. The *Select Query* pane is used to build the desired query. Queries are built out of selection of "contexts," currently including diagnosis, brain region of interest, gender, age, and species. Diagnoses selection is implemented in a drop down list format. Similarly, brain region of interest can also be chosen from a drop down list or by selecting it in the clickable atlas provided below the search pane. The specific age range can be provided by selecting Young (0–9 years), Adolescent (10–18 years), Young



About the portal

CANDI lab

University of Massachusetts Medical School

NITRC

Contact us

Brain Structures: Intracranial
Or choose ROI by clicking on the atlas below.

Hemisphere: Any

Gender: Any

Age range: All 0 to 150 years.
You can modify the age range as per your preference.

Species: All

search Reset search

Current Query:

Diagnosis:			
Brain Structures:		Hemisphere:	
Gender:		Species:	
Age Range:			

Matched Resources:

Please select the search options.

Regions of interest (ROI) can be selected either by clicking on the atlas on the right or using the drop down menu.

You can then either keep the suggested hemisphere or click 'Cancel' to get hemisphere independent results.

This portal is an effort to develop knowledge environment wrapper that provides organization and integration of multiple information sources. Organized semantically as a Structure-Centered Database, following the conceptual design like introduced in the 'cell-centered database' of Martone, the portal will group information sources by context, and permit the user to interactively 'narrow' or 'broaden' the scope of the information resources that are available and relevant to the specific context.

> Choose query parameters and click the 'Search' button.

> Bring the mouse over a link to display the results in this pane.

> Click the link to display the results in a new tab.

* Please note that IBVD reports the standard deviation (SD) of the volumes, hence, if an article reports standard error (SE), the result is entered as SD = SE x square_root(number of subjects).

FIGURE 1 | The four pane window format. The portal has a four-pane format consisting of Select Query pane, Clickable brain atlas pane, Resource match pane and the Results pane. The Research match pane displays any matching information and data resources. The results for each of the matched resources can either be viewed in the Results pane or be opened in a separate tab.

and Adolescent (0–18 years), or Adult (19–150 years). The minimum and the maximum bounds of the age range can be further modified by entering the values directly.

The *Anatomic Atlas* pane supports the user selection of the anatomic context for the *Select Query* pane. This is accomplished through the use of the canvas feature of HTML5. The atlas itself is based on FreeSurfer segmented structural MRI scan of a typically developing 15-year-old female subject. The user can navigate between coronal slices and select regions by mouse click.

The *Resource Match* pane displays links to various available data and information resources, the output for which can be viewed either in the *Results* pane or in a separate web-browser tab. For the data resources a summary of the numbers of datasets available per resource is provided.

RESOURCES

A specific set of remote resources is currently supported which are queried using the public web services. We make a distinction between two types of resources: information resources and MRI data resources. As will be elaborated upon below, these two classes of resource, and the results returned, require different handling. The following is the set of resources that are currently included:

Information Resources:

PubMed: Biomedical literature from MEDLINE, life science journals, and online books).

Entrez Gene: Genetic records including nomenclature, reference sequences, maps, pathways, variations, phenotypes, etc.

IBVD (Kennedy et al., 2003): Internet Brain Volume Database (IBVD) is a database of volumetric information of different brain structures from over 600 publications and over 15 thousand individual volumes.

PubBrain (Kalar et al., 2007): A meta-analysis tool providing numerical and pictorial representation of prevalence of brain structure bibliographic references as identified by the query terms found in PubMed.

MRI Data Resources:

CANDIShare (Kennedy et al., 2012): MRI datasets of structural brain images, as well as their anatomic segmentations, demographic, and behavioral data and a set of related morphometric resources for young and adolescent typically developing and psychiatric disorder populations.

OASIS datasets on XNAT Central: MRI datasets of very mild to moderate Alzheimer's disease patients including demented and non-demented subjects as well as normal controls between the ages of 18 and 96 years in the cross-sectional dataset, and between the ages of 60 and 96 years in the longitudinal dataset where the subjects are scanned over two or more visits.

**OASIS-brains Database (Marcus et al., 2007):* OASIS datasets are available through www.oasis-brains.org with additional demographic details such as gender, grouping into demented/non-demented groups and CDR scores (unavailable for download through the XNAT central OASIS release).

fCON1000 (Biswal et al., 2010): Neuroimaging database of resting-state functional magnetic resonance imaging data of healthy subjects.

**PING (Brown et al., 2012):* Large MRI and genetics data set of typically developing children between the ages of 3 and 20 years.

**NIH_PD (Evans, 2006):* NIH Pediatric database (NIH_PD) of longitudinal MRI data of typically developing children and adolescents scanned during three visits.

**ADHD200 (Fair et al., 2012):* Publically released dataset of resting-state fMRI and anatomical imaging for 491 typically developing individuals, and 285 in children and adolescents with ADHD between the ages of 7 and 21 years.

**ABIDE (Di Martino et al., 2013):* Autism Brain Imaging Data Exchange (ABIDE) dataset contains resting state functional imaging and morphometric data from 539 individuals with autism spectrum disorder and 573 typical controls.

Resources marked with * require some sort of user registration process in order to access the imaging data. While the portal provides simple indication of the types of data that would be obtained with the query (in terms of subjects matching age, gender, and diagnostic characteristics) users are required to acquire their own specific access authentication.

OPERATION

Once the user fills in their query terms and clicks the submit button in the "Select Query" pane, resources matching the query are displayed in the Resource Match pane. Each of the information resources can then either be viewed in the Result pane, or opened in a separate tab. For all the imaging databases, the demographic information of available data is displayed in the result pane and the user is directed to the respective websites in order to complete any necessary registration process in order to download the data.

When queries are run against IBVD, CANDIShare, OASIS, fCON1000, PING, NIH_PD, ADHD200, or ABIDE with diagnosis included, the implication is that the user is interested in the contrasts between "typical" and this diagnosis. Therefore, while running the queries on these resources, the query is conducted twice, once for the diagnosis and other context qualifiers, and additionally for age and gender matched normal controls.

Also, data returned from specific resources can be processed locally to derive additional representations of that data. Specific examples of this include automated provision of a z-score table and a z-score plot for the ROI volumes returned from the IBVD results, and a generation of the top five most published genes listing for the Entrez gene results for any given disorder of interest query.

IMPLEMENTATION

The portal is designed as a stand-alone application. Instead of downloading this application to each user, the application is hosted on a publically available computer and accessed via web-based browser. HTML5 is used to develop the user interface. Dynamic functionality is implemented using JavaScript. The point-and-click brain atlas is implemented using the canvas feature of HTML5. The atlas itself is based on FreeSurfer segmented structural MRI scan of a normal 15-year-old female subject.

In the absence of a standard API that facilitates interoperation with all neuroscience resources, we maintain a resource-by-resource catalog of queryable terms and the context that these terms are pertinent to. When queries are implemented we maintain a resource-specific specification of each queryable item and the syntax of the query for that resource. Given the variations between the different resources, the query for each resource is generated independently. **Figure 2** provides a pictorial view of how different resources are queried. Either all or a subset of the search criteria is used to generate the query for an individual resource. e.g., PubMed results are based on the diagnosis, brain region, hemisphere, gender, age range as chosen by description (young, adolescent, young, and adolescent, or adult), and species queried, where as PubBrain results are purely based on the diagnosis, gender and age range in years. This approach provides modularity to the portal, making it easier to modify the current queries or add any new resources in future. Another advantage of this approach is that the data is presented in a way that would be most useful to the user. For example, though the IBVD results are limited by age range, we provide a IBVD based z-score plot over the entire age range from young to adult, hence giving the user an overview of changes in volumes of the ROI as a factor of age.

CANDIShare, fCON, ABIDE, ADHD200, and OASIS datasets available through XNAT are queried using Python and pyxnat (Schwartz et al., 2012). NIH pediatric database and PING database are currently not available for direct web query. The results for these resources are made available to the user by querying the demographics available to us. Currently each resource query is custom created (e.g., for IBVD the age range is inputted in the form of minimum and maximum age as opposed to PubMed for which either of young, adolescent or old is used). The results frame utilizes the inline frame feature, hence enabling display of various resource webpages in the same window.

LOCAL DATA MANIPULATION

As indicated above, the portal supports a layer of local analyses that can be inserted to process or condition the results of each of the queries to add information or contest. We demonstrate two examples of this local processing that enhance the interpretational value of information returned from the query in support of the clinical end user.

First, the anatomic volume results from IBVD query for a disorder of interest and healthy control groups are represented as a flat table and a graph of raw volumes. A common way of interpreting these results would be for the end user to further parse these results to find matching disorder—normal control results pairs that originate from the same article. These results can then be converted in to a z-score, which is a ratio of difference between the disorder and control group mean to the average standard deviation of the two groups, using the formula stated below. Any articles that might have multiple disorder-normal pairs that can't be separated using gender or hemisphere information are marked as multiple matches. The results are provided to the user as a table as well as a z-score vs. age plot. This sequence of data

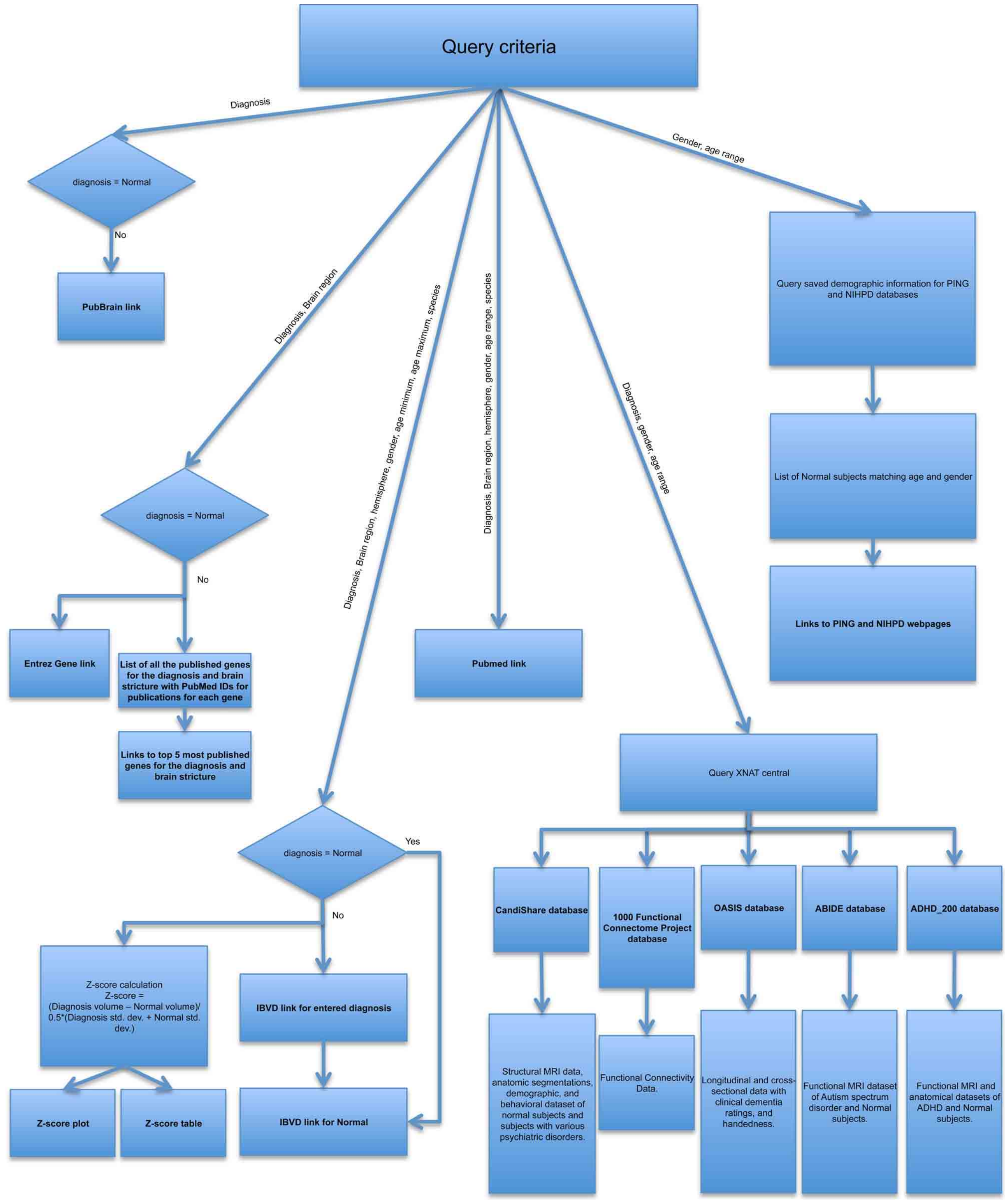

Query criteria
Diagnosis
diagnosis = Normal
No
PubBrain link
Diagnosis, Brain region
diagnosis = Normal
No
Entrez Gene link
List of all the published genes for the diagnosis and brain stricture with PubMed IDs for publications for each gene
Links to top 5 most published genes for the diagnosis and brain stricture
Diagnosis, Brain region, hemisphere, gender, age minimum, age maximum, species
diagnosis = Normal
Yes
No
Z-score calculation
Z-score =
(Diagnosis volume – Normal volume)/ 0.5*(Diagnosis std. dev. + Normal std. dev.)
Z-score plot
Z-score table
IBVD link for entered diagnosis
IBVD link for Normal
Diagnosis, Brain region, hemisphere, gender, age range, species
Pubmed link
Diagnosis, gender, age range
Query XNAT central
CandiShare database
Structural MRI data, anatomic segmentations, demographic, and behavioral dataset of normal subjects and subjects with various psychiatric disorders.
1000 Functional Connectome Project database
Functional Connectivity Data.
OASIS database
Longitudinal and cross-sectional data with clinical dementia ratings, and handedness.
ABIDE database
Functional MRI dataset of Autism spectrum disorder and Normal subjects.
ADHD_200 database
Functional MRI and anatomical datasets of ADHD and Normal subjects.
Gender, age range
Query saved demographic information for PING and NIHPD databases
List of Normal subjects matching age and gender
Links to PING and NIHPD webpages

FIGURE 2 | Flowchart depicting how the query input is tailored to requirements of various resources.

interpretation steps is automated in the portal in order to provide the end user an added context to the results that are returned. The multiple-matched results are not included in the plot.

$$\text{Z-score} = \frac{\left(\begin{array}{r}\text{disorder group mean volume}-\\ \text{control group mean volume}\end{array}\right)}{\left[\begin{array}{c}0.5 \times (\text{disorder volume std} +\\ \text{control volume std})\end{array}\right]}$$

Finally, a trend-line is generated for the z-score vs. average age plot using the locally weighted scatterplot smoothing (lowess) non-parametric regression (Cleveland, 1979; Cleveland and Devlin, 1988).

As a second example of local result manipulation, we consider the Entrez gene database query result. Initially, this query provides a list of associated genes that is ordered relative to last update of their Entrez gene record (such that the most recently published gene on top of the list). However, as the list of genes returned from a query becomes large, recency of record update is not the optimum criterion for identifying the most salient genetic implications. In this case, the portal will run a process that takes these results and rank order organizes it with respect to the number of publications per gene for the query. The top five of the most published genes are presented under the Gene tab along with a list of all the genes published for that disorder—ROI combination and PubMed IDs of publications for each gene.

USE CASES

We illustrate the query building functionality through examination of the details for a sample query:

Disorder: Bipolar Disorder
Brain Structure: Amygdala
Hemisphere: Left
Gender: Female
Age: Young or 0–9 years
Species: Human

For each resource, the following table shows the mapping of the context terms to the actual query.

Resource	Diagnosis	Brain structure	Hemisphere	Gender	Age range	Species
PubMed	Bipolar disorder[a]	Amygdala	–	Female	Young	Human
EntrezGene	Bipolar disorder	Amygdala	–	–	–	–
IBVD	Bipolar disorder	Amygdala	Left	Female	Age min = 0 Age Max = 9	Human
	Normal	Amygdala	Left	Female	Age min = 0 Age Max = 9	Human
PubBrain	Bipolar disorder	–	–	–	–	–
CANDIShare	Bipolar disorder	-	-	F*	Age min = 0 Age Max = 9	–
	Normal[b]	–	–	F*	Age min = 0 Age Max = 9	–
OASIS datasets on XNAT	–**	–	–	–	Age min = 0 Age Max = 9	–
OASIS-brains database	CDR score =" " or CDR score = 0	–	–	F*	Age min = 0 Age Max = 9	–
fCON1000	Normal	–	–	F*	Age min = 0 Age Max = 9	–
PING	Normal[#]	–	–	F*	Age min = 0 Age Max = 9	–
NIH_PD	–[##]	–	–	F*	Age min = 0 Age Max = 9	–
ADHD200	Control[b]	–	–	F*	Age min = 0 Age Max = 9	–
ABIDE	Typically developing[b]	–	–	F*	Age min = 0 Age Max = 9	–

[a] *Databases such as PubMed expand individual search criterion to match their own terminology.*

[b] *Based on a given resource, the search parameters are modified to fit the reported diagnosis, such as "Typically developing" for ABIDE, "Control" for ADHD200, and "Normal" for CANDIShare.*

F for female is compared with the capitalized first letter of the reported gender to determine a match.

**Since diagnosis is not available as a part of OASIS dataset demographics on XNAT, it is not queried.*

[#] *PING dataset limits its subjects to those without any confirmed diagnosis of autism, mental retardation, bipolar disorder, schizophrenia, or any neurological disorder such as cerebral palsy, fetal alcohol syndrome, Down's syndrome, fragile X, cerebral neoplasm, bacterial meningitis, epilepsy, and hence the subjects are considered "Typically developing."*

[##] *NIH_PD exclusion criteria included diagnosis for any major medical illness, congenital abnormalities, heart problems, cancer, lead poisoning, seizures, CNS Infection, head injury, significant hearing loss, language disorder, mood disorder, Conduct, AD/HD, Tic, Eating disorders, as well as, presence of bipolar disorder, chronic depression, psychotic, AD/HD, drug dependence, or PDD in first degree relatives. Hence, can be considered typically developing and no diagnosis is reported.*

USER ACCESS

The portal is freely accessible as a website hosted at http://childportal.virtualbrain.org. This host is an Amazon EC2 NITRC-Computational Environment Ubuntu 12.04 platform. The underlying computational power of the EC2 instance can be scaled to meet variations in portal demand.

RESULTS

The operation of the portal is best illustrated through a number of examples/case studies. **Figure 3** provides an overview and comparison of the results of two different contextual queries. The left hand column displays the results for a query on "Diagnosis: ADHD; Brain Structure: cerebrum; and Age: Adolescent." The right column displays query for "Diagnosis: Bipolar Disorder; Brain Structure: Amygdala; Gender: female; and Age: Young" (**Figure 3A**). The links are generated for each resource and presented to the user. The specific, query-dependent version for each resource is displayed. As shown in **Figure 3B**, the IBVD results for the query are presented along with z-score table and plot for the disorder-ROI combination. Similarly, the Entrez Gene results are further processed and presented with the top five most published genes for the disorder-ROI combination, as shown in **Figure 3D** for the Bipolar-Amygdala query which has only four genes that actually have common publications for Bipolar disorder and Amygdala despite the list of 33 genes produced by Entrez Gene. The user can further manipulate the output for each of the information resource, if necessary. For example, Entrez gene did not have any entries for the combined query of ADHD and cerebrum (**Figure 3D**). The user can in such cases modify the search in the results pane for that particular resource only to look at gene entries for ADHD alone.

For the MRI data resources, the available data for the disorder as well as normal controls are displayed for the age range in question. If any resource has not specified any disorder in their demographics, those results are displayed as well and listed as "unspecified disorder." Since many data resources require a user to register with them before the data can be released, the portal points them to the resource websites in case they want to access the data.

DISCUSSION

Despite the presence of numerous neuroinformatics resources that are available to the clinician, we believe that the Child Psychiatry Portal is the first effort to create a platform to consolidate these data and information resources specifically for the needs of the pediatric psychiatry researcher. Currently the target resources include IBVD, PubMed, Entrez gene, and PubBrain, NIH pediatric database, PING, CANDIShare, 1000 Functional Connectomes Project (FCON), and OASIS longitudinal and cross-sectional studies. We list data that is available through the five resources we query whether it matches the entire query as specified, including diagnosis, or it matches the age range and gender characteristics specified for control (typically developing) subject data. The power of this approach comes not through the complexity of any one query, but rather the collection and integration of a wide variety of resource queries under one application where most of the operational details and idiosyncrasies of the of the individual resources can be initially abstracted away from the end user. Ultimately, use of these varied resources by users not intimately trained in the details of each site will be critical to wide-spread utilization of these many valued data sources.

ADVANTAGES

The first and foremost advantage of this portal is that it brings information from multiple complex resources into a simple single uniform framework without requiring myriad of resource-specific syntax knowledge. Additionally, the portal has an extensible modular architecture. Hence, it can be expanded to include results from any additional resources as and when they become available. We understand that a user might already be aware of some of the individual resources and be well versed in navigating these resources. However, the ability to modify the search parameters for multiple resources simultaneously should be an advantage to the user over having to go through each of the resources individually. This represents a dramatic saving in terms of time in order to look for availability of information from multiple resources every time one needs to modify a search parameter. See the Supplementary Material for the description of performing query on individual resources presented in the portal without the portal.

A second critical advantage of the system is to introduce a data manipulation layer between the raw results from the various resources and the presentation of this information in a form that is best suited to the end user. Databases that do a good job of collecting data cannot anticipate and support every re-use and re-interpretation that can be envisioned for their data. As users develop convenient ways to interpret data, there need to be equally convenient ways to implement and disseminate these views to the end users that are more flexible than building upon the database infrastructure itself.

Finally, within the neuroimaging search functionality, acknowledging that multiple data sources (implemented using multiple data hosting platforms) will always exist and that the content from these sources will ultimately need to be pooled, requires the development of a "higher-order" search platform that can span a dynamically changing landscape of image data resources. The ability to both quickly and efficiently integrate data sets between sources and discover the presence of additional data sources will grow in importance as the amount of shared image data, number of providers, and variety of access terms increases.

LIMITATIONS

As it can be seen from the results, though numerous studies are published every year, a very small portion of the data is made available and it is further limited in case of studies of psychiatric disorders in children. This highlights the need to promote data sharing to researchers. Currently only a limited number of MRI Data resources are available for downloading patient related imaging data. Despite this, the user can at least take advantage of any available control data, perhaps for integration with their own patient datasets.

Another important thing to note is, not all available data resources follow similar rules for nomenclature. For example, the

A

Search criteria

Diagnosis: ADHD

Brain Structure: Cerebrum

Age: Adolescent (10 to 18 years)

Diagnosis: Bipolar Disorder

Brain Structure: Amygdala

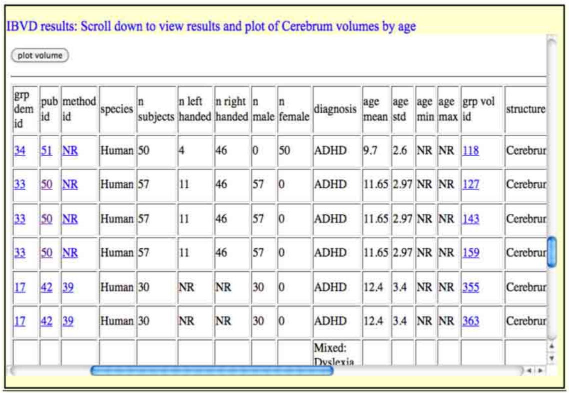
Gender: Female

Age: Young (0 to 9 years)

B

IBVD results

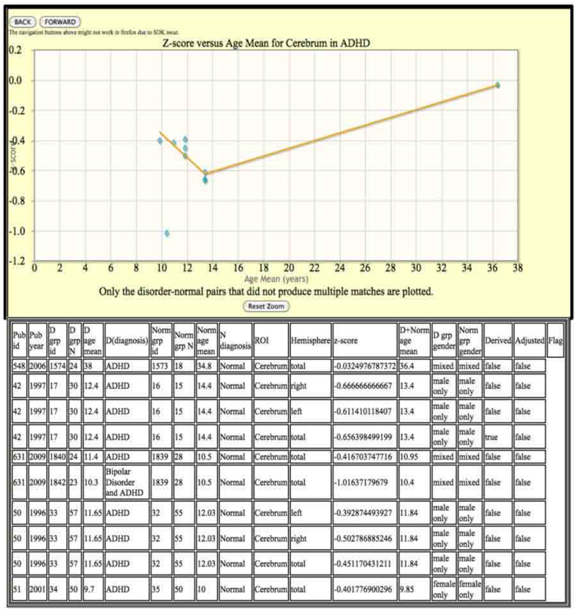
Cerebrum volume in ADHD has been studied in 16 records from 7 publication.



IBVD results: Scroll down to view results and plot of Cerebrum volumes by age

grp dem id	pub id	method id	species	n subjects	n left handed	n right handed	n male	n female	diagnosis	age mean	age std	age min	age max	grp vol id	structure
34	51	NR	Human	50	4	46	0	50	ADHD	9.7	2.6	NR	NR	118	Cerebrur
33	50	NR	Human	57	11	46	57	0	ADHD	11.65	2.97	NR	NR	127	Cerebrur
33	50	NR	Human	57	11	46	57	0	ADHD	11.65	2.97	NR	NR	143	Cerebrur
33	50	NR	Human	57	11	46	57	0	ADHD	11.65	2.97	NR	NR	159	Cerebrur
17	42	39	Human	30	NR	NR	30	0	ADHD	12.4	3.4	NR	NR	355	Cerebrur
17	42	39	Human	30	NR	NR	30	0	ADHD	12.4	3.4	NR	NR	363	Cerebrur
									Mixed: Dyslexia						

The z-score plot and table of resulting ADHD cerebrum records

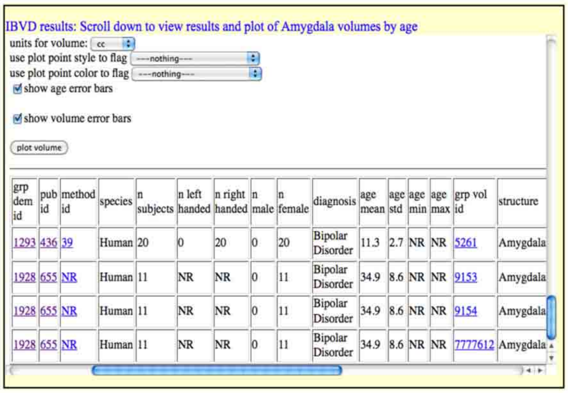


BACK
FORWARD
The navigation buttons above might not work in firefox due to SDK issue.
Z-score versus Age Mean for Cerebrum in ADHD
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.2
Z-score
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
Age Mean (years)
Reset Zoom

Only the disorder-normal pairs that did not produce multiple matches are plotted.

Pub id	Pub year	D grp id	D grp N	D age mean	D(diagnosis)	Norm grp id	Norm grp N	Norm age mean	N diagnosis	ROI	Hemisphere	z-score	D+Norm age mean	D grp gender	Norm grp gender	Derived	Adjusted	Flag
548	2006	1574	24	38	ADHD	1573	18	34.8	Normal	Cerebrum	total	-0.0324976787372	36.4	mixed	mixed	false	false	
42	1997	17	30	12.4	ADHD	16	15	14.4	Normal	Cerebrum	right	-0.666666666667	13.4	male only	male only	false	false	
42	1997	17	30	12.4	ADHD	16	15	14.4	Normal	Cerebrum	left	-0.611410118407	13.4	male only	male only	false	false	
42	1997	17	30	12.4	ADHD	16	15	14.4	Normal	Cerebrum	total	-0.656398499199	13.4	male only	male only	true	false	
631	2009	1840	24	11.4	ADHD	1839	28	10.5	Normal	Cerebrum	total	-0.416703747716	10.95	mixed	mixed	false	false	
631	2009	1842	23	10.3	Bipolar Disorder and ADHD	1839	28	10.5	Normal	Cerebrum	total	-1.01637179679	10.4	mixed	mixed	false	false	
50	1996	33	57	11.65	ADHD	32	55	12.03	Normal	Cerebrum	left	-0.392874493927	11.84	male only	male only	false	false	
50	1996	33	57	11.65	ADHD	32	55	12.03	Normal	Cerebrum	right	-0.502786885246	11.84	male only	male only	false	false	
50	1996	33	57	11.65	ADHD	32	55	12.03	Normal	Cerebrum	total	-0.451170431211	11.84	male only	male only	false	false	
51	2001	34	50	9.7	ADHD	35	50	10	Normal	Cerebrum	total	-0.401776900296	9.85	female only	female only	false	false	

Bipolar disorder + Amygdala + female = only 2 publications and 4 total entries.



IBVD results: Scroll down to view results and plot of Amygdala volumes by age

units for volume: cc

use plot point style to flag ---nothing---

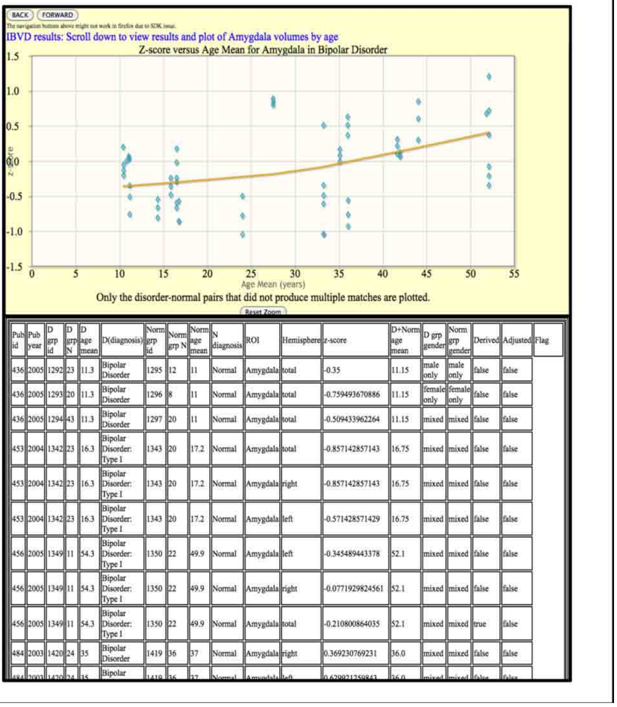
use plot point color to flag ---nothing---

☑ show age error bars

☑ show volume error bars

grp dem id	pub id	method id	species	n subjects	n left handed	n right handed	n male	n female	diagnosis	age mean	age std	age min	age max	grp vol id	structure
1293	436	39	Human	20	0	20	0	20	Bipolar Disorder	11.3	2.7	NR	NR	5261	Amygdala
1928	655	NR	Human	11	NR	NR	0	11	Bipolar Disorder	34.9	8.6	NR	NR	9153	Amygdala
1928	655	NR	Human	11	NR	NR	0	11	Bipolar Disorder	34.9	8.6	NR	NR	9154	Amygdala
1928	655	NR	Human	11	NR	NR	0	11	Bipolar Disorder	34.9	8.6	NR	NR	7777612	Amygdala

The z-score plot and table of resulting Bipolar Amygdala records

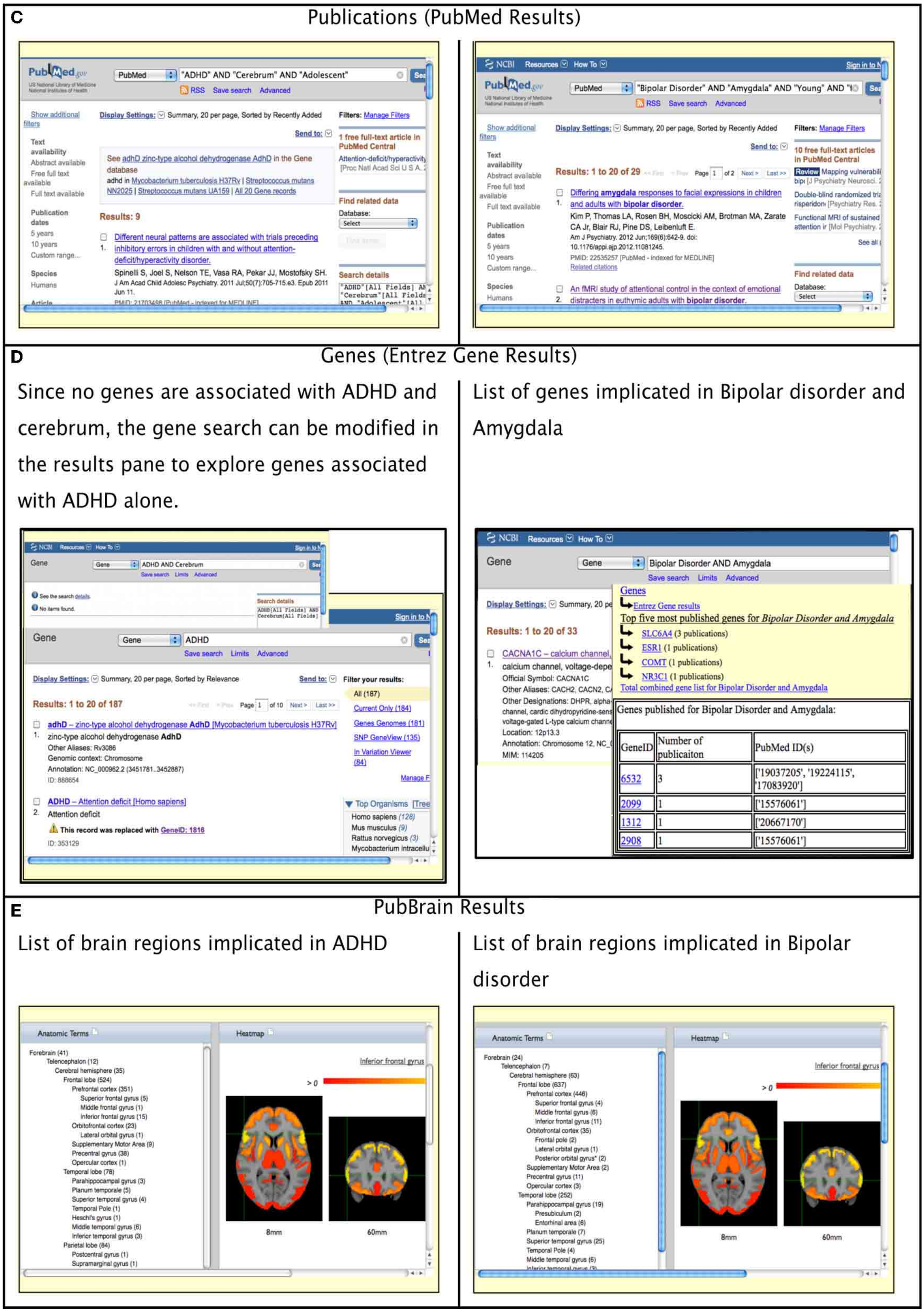


BACK
FORWARD
The navigation buttons above might not work in firefox due to SDK issue.
IBVD results: Scroll down to view results and plot of Amygdala volumes by age
Z-score versus Age Mean for Amygdala in Bipolar Disorder
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
Z-score
0
5
10
15
20
25
30
35
40
45
50
55
Age Mean (years)
Reset Zoom

Only the disorder-normal pairs that did not produce multiple matches are plotted.

Pub id	Pub year	D grp id	D grp N	D age mean	D(diagnosis)	Norm grp id	Norm grp N	Norm age mean	N diagnosis	ROI	Hemisphere	z-score	D+Norm age mean	D grp gender	Norm grp gender	Derived	Adjusted	Flag
436	2005	1292	23	11.3	Bipolar Disorder	1295	12	11	Normal	Amygdala	total	-0.35	11.15	male only	male only	false	false	
436	2005	1293	20	11.3	Bipolar Disorder	1296	8	11	Normal	Amygdala	total	-0.759493670886	11.15	female only	female only	false	false	
436	2005	1294	43	11.3	Bipolar Disorder	1297	20	11	Normal	Amygdala	total	-0.509433962264	11.15	mixed	mixed	false	false	
453	2004	1342	23	16.3	Bipolar Disorder: Type I	1343	20	17.2	Normal	Amygdala	total	-0.857142857143	16.75	mixed	mixed	false	false	
453	2004	1342	23	16.3	Bipolar Disorder: Type I	1343	20	17.2	Normal	Amygdala	right	-0.857142857143	16.75	mixed	mixed	false	false	
453	2004	1342	23	16.3	Bipolar Disorder: Type I	1343	20	17.2	Normal	Amygdala	left	-0.571428571429	16.75	mixed	mixed	false	false	
456	2005	1349	11	54.3	Bipolar Disorder: Type I	1350	22	49.9	Normal	Amygdala	left	-0.345489443378	52.1	mixed	mixed	false	false	
456	2005	1349	11	54.3	Bipolar Disorder: Type I	1350	22	49.9	Normal	Amygdala	right	-0.0771929824561	52.1	mixed	mixed	false	false	
456	2005	1349	11	54.3	Bipolar Disorder: Type I	1350	22	49.9	Normal	Amygdala	total	-0.210800864035	52.1	mixed	mixed	true	false	
484	2003	1420	24	35	Bipolar Disorder	1419	36	37	Normal	Amygdala	right	0.369230769231	36.0	mixed	mixed	false	false	
484	2003	1420	24	35	Bipolar	1419	36	37	Normal	Amygdala	left	0.629921259843	36.0	mixed	mixed	false	false	

FIGURE 3 | Continued



C Publications (PubMed Results)

PubMed | "ADHD" AND "Cerebrum" AND "Adolescent"

RSS Save search Advanced

Show additional filters

Text availability
Abstract available
Free full text available
Full text available

Publication dates
5 years
10 years
Custom range...

Species
Humans

Display Settings: Summary, 20 per page, Sorted by Recently Added

Send to:

See adhD zinc-type alcohol dehydrogenase AdhD in the Gene database
adhd in Mycobacterium tuberculosis H37Rv | Streptococcus mutans NN2025 | Streptococcus mutans UA159 | All 20 Gene records

Results: 9

□ 1. Different neural patterns are associated with trials preceding inhibitory errors in children with and without attention-deficit/hyperactivity disorder.
Spinelli S, Joel S, Nelson TE, Vasa RA, Pekar JJ, Mostofsky SH.
J Am Acad Child Adolesc Psychiatry. 2011 Jul;50(7):705-715.e3. Epub 2011 Jun 11.
PMID: 21703498 [PubMed - indexed for MEDLINE]

Filters: Manage Filters

1 free full-text article in PubMed Central
Attention-deficit/hyperactivity [Proc Natl Acad Sci U S A.

Find related data
Database: Select

Search details
"ADHD"[All Fields] AND "Cerebrum"[All Fields] AND "Adolescent"[All

NCBI Resources How To Sign in to N

PubMed | "Bipolar Disorder" AND "Amygdala" AND "Young" AND "I

RSS Save search Advanced

Show additional filters

Text availability
Abstract available
Free full text available
Full text available

Publication dates
5 years
10 years
Custom range...

Species
Humans

Display Settings: Summary, 20 per page, Sorted by Recently Added

Send to:

Results: 1 to 20 of 29 << First < Prev Page 1 of 2 Next > Last >>

□ 1. Differing **amygdala** responses to facial expressions in children and adults with **bipolar disorder**.
Kim P, Thomas LA, Rosen BH, Moscicki AM, Brotman MA, Zarate CA Jr, Blair RJ, Pine DS, Leibenluft E.
Am J Psychiatry. 2012 Jun;169(6):642-9. doi: 10.1176/appi.ajp.2012.11081245.
PMID: 22535257 [PubMed - indexed for MEDLINE]
Related citations

□ 2. An fMRI study of attentional control in the context of emotional distracters in euthymic adults with **bipolar disorder**.

Filters: Manage Filters

10 free full-text articles in PubMed Central
Review Mapping vulnerabili bip [J Psychiatry Neurosci.
Double-blind randomized tri risperidon [Psychiatry Res.
Functional MRI of sustained attention ir [Mol Psychiatry.
See all

Find related data
Database: Select

D Genes (Entrez Gene Results)

Since no genes are associated with ADHD and cerebrum, the gene search can be modified in the results pane to explore genes associated with ADHD alone.

NCBI Resources How To Sign in to N

Gene | Gene | ADHD AND Cerebrum

Save search Limits Advanced

See the search details.
No items found.

Search details
ADHD[All Fields] AND Cerebrum[All Fields]

Sign in to N

Gene | Gene | ADHD

Save search Limits Advanced

Display Settings: Summary, 20 per page, Sorted by Relevance

Send to:

Results: 1 to 20 of 187 << First < Prev Page 1 of 10 Next > Last >>

□ 1. **adhD** – zinc-type alcohol dehydrogenase **AdhD** [Mycobacterium tuberculosis H37Rv]
zinc-type alcohol dehydrogenase **AdhD**
Other Aliases: Rv3086
Genomic context: Chromosome
Annotation: NC_000962.2 (3451781..3452887)
ID: 888654

□ 2. **ADHD** – Attention deficit [Homo sapiens]
Attention deficit
⚠ **This record was replaced with GeneID: 1816**
ID: 353129

Filter your results:
All (187)
Current Only (184)
Genes Genomes (181)
SNP GeneView (135)
In Variation Viewer (84)
Manage F

▼ Top Organisms [Tree
Homo sapiens (128)
Mus musculus (9)
Rattus norvegicus (3)
Mycobacterium intracellu

List of genes implicated in Bipolar disorder and Amygdala

NCBI Resources How To

Gene | Gene | Bipolar Disorder AND Amygdala

Save search Limits Advanced

Display Settings: Summary, 20 pe

Results: 1 to 33 of 33

□ 1. CACNA1C – calcium channel,
calcium channel, voltage-depe
Official Symbol: CACNA1C
Other Aliases: CACH2, CACN2, C
Other Designations: DHPR, alpha channel, cardic dihydropyridine-sens voltage-gated L-type calcium channe
Location: 12p13.3
Annotation: Chromosome 12, NC_
MIM: 114205

Genes
Entrez Gene results
Top five most published genes for *Bipolar Disorder and Amygdala*
SLC6A4 (3 publications)
ESR1 (1 publications)
COMT (1 publications)
NR3C1 (1 publications)
Total combined gene list for Bipolar Disorder and Amygdala

Genes published for Bipolar Disorder and Amygdala:

GeneID	Number of publicaiton	PubMed ID(s)
6532	3	['19037205', '19224115', '17083920']
2099	1	['15576061']
1312	1	['20667170']
2908	1	['15576061']

E PubBrain Results

List of brain regions implicated in ADHD

Anatomic Terms

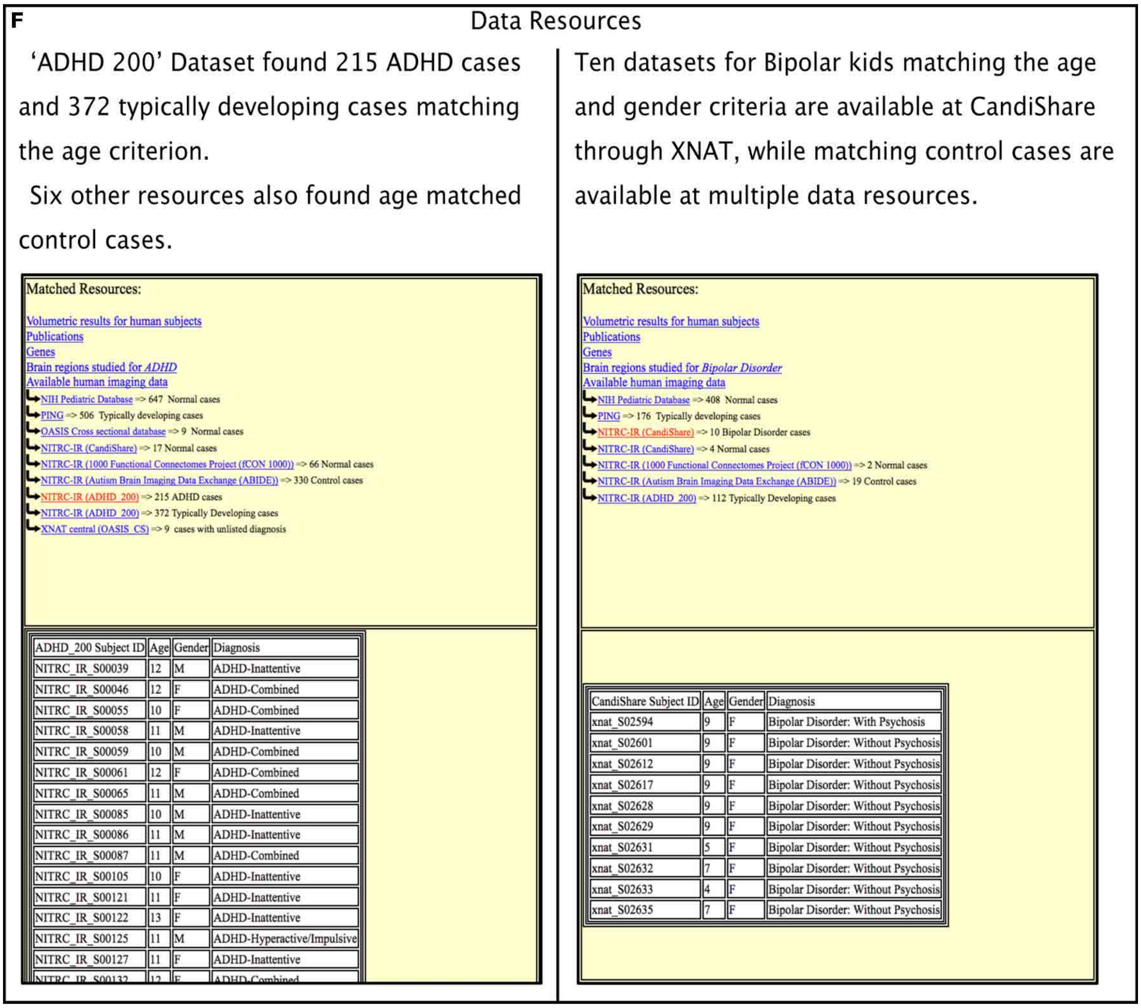
Forebrain (41)
- Telencephalon (12)
  - Cerebral hemisphere (35)
    - Frontal lobe (524)
      - Prefrontal cortex (351)
        - Superior frontal gyrus (5)
        - Middle frontal gyrus (1)
        - Inferior frontal gyrus (15)
      - Orbitofrontal cortex (23)
        - Lateral orbital gyrus (1)
      - Supplementary Motor Area (9)
      - Precentral gyrus (38)
      - Opercular cortex (1)
    - Temporal lobe (78)
      - Parahippocampal gyrus (3)
      - Planum temporale (5)
      - Superior temporal gyrus (4)
      - Temporal Pole (1)
      - Heschl's gyrus (1)
      - Middle temporal gyrus (6)
      - Inferior temporal gyrus (3)
    - Parietal lobe (84)
      - Postcentral gyrus (1)
      - Supramarginal gyrus (1)

List of brain regions implicated in Bipolar disorder

Anatomic Terms

Forebrain (24)
- Telencephalon (7)
  - Cerebral hemisphere (63)
    - Frontal lobe (637)
      - Prefrontal cortex (446)
        - Superior frontal gyrus (4)
        - Middle frontal gyrus (6)
        - Inferior frontal gyrus (11)
      - Orbitofrontal cortex (35)
        - Frontal pole (2)
        - Lateral orbital gyrus (1)
        - Posterior orbital gyrus* (2)
      - Supplementary Motor Area (2)
      - Precentral gyrus (11)
      - Opercular cortex (3)
    - Temporal lobe (252)
      - Parahippocampal gyrus (19)
        - Presubiculum (2)
        - Entorhinal area (6)
      - Planum temporale (7)
      - Superior temporal gyrus (25)
      - Temporal Pole (4)
      - Middle temporal gyrus (6)
      - Inferior temporal gyrus (3)

FIGURE 3 | Continued



Data Resources

'ADHD 200' Dataset found 215 ADHD cases and 372 typically developing cases matching the age criterion.

Six other resources also found age matched control cases.

Matched Resources:

Volumetric results for human subjects
Publications
Genes
Brain regions studied for *ADHD*
Available human imaging data
↳NIH Pediatric Database => 647 Normal cases
↳PING => 506 Typically developing cases
↳OASIS Cross sectional database => 9 Normal cases
↳NITRC-IR (CandiShare) => 17 Normal cases
↳NITRC-IR (1000 Functional Connectomes Project (fCON 1000)) => 66 Normal cases
↳NITRC-IR (Autism Brain Imaging Data Exchange (ABIDE)) => 330 Control cases
↳NITRC-IR (ADHD_200) => 215 ADHD cases
↳NITRC-IR (ADHD_200) => 372 Typically Developing cases
↳XNAT central (OASIS_CS) => 9 cases with unlisted diagnosis

ADHD_200 Subject ID	Age	Gender	Diagnosis
NITRC_IR_S00039	12	M	ADHD-Inattentive
NITRC_IR_S00046	12	F	ADHD-Combined
NITRC_IR_S00055	10	F	ADHD-Combined
NITRC_IR_S00058	11	M	ADHD-Inattentive
NITRC_IR_S00059	10	M	ADHD-Combined
NITRC_IR_S00061	12	F	ADHD-Combined
NITRC_IR_S00065	11	M	ADHD-Combined
NITRC_IR_S00085	10	M	ADHD-Inattentive
NITRC_IR_S00086	11	M	ADHD-Inattentive
NITRC_IR_S00087	11	M	ADHD-Combined
NITRC_IR_S00105	10	F	ADHD-Inattentive
NITRC_IR_S00121	11	F	ADHD-Inattentive
NITRC_IR_S00122	13	F	ADHD-Inattentive
NITRC_IR_S00125	11	M	ADHD-Hyperactive/Impulsive
NITRC_IR_S00127	11	F	ADHD-Inattentive
NITRC_IR_S00132	12	F	ADHD-Combined

Ten datasets for Bipolar kids matching the age and gender criteria are available at CandiShare through XNAT, while matching control cases are available at multiple data resources.

Matched Resources:

Volumetric results for human subjects
Publications
Genes
Brain regions studied for *Bipolar Disorder*
Available human imaging data
↳NIH Pediatric Database => 408 Normal cases
↳PING => 176 Typically developing cases
↳NITRC-IR (CandiShare) => 10 Bipolar Disorder cases
↳NITRC-IR (CandiShare) => 4 Normal cases
↳NITRC-IR (1000 Functional Connectomes Project (fCON 1000)) => 2 Normal cases
↳NITRC-IR (Autism Brain Imaging Data Exchange (ABIDE)) => 19 Control cases
↳NITRC-IR (ADHD_200) => 112 Typically Developing cases

CandiShare Subject ID	Age	Gender	Diagnosis
xnat_S02594	9	F	Bipolar Disorder: With Psychosis
xnat_S02601	9	F	Bipolar Disorder: Without Psychosis
xnat_S02612	9	F	Bipolar Disorder: Without Psychosis
xnat_S02617	9	F	Bipolar Disorder: Without Psychosis
xnat_S02628	9	F	Bipolar Disorder: Without Psychosis
xnat_S02629	9	F	Bipolar Disorder: Without Psychosis
xnat_S02631	5	F	Bipolar Disorder: Without Psychosis
xnat_S02632	7	F	Bipolar Disorder: Without Psychosis
xnat_S02633	4	F	Bipolar Disorder: Without Psychosis
xnat_S02635	7	F	Bipolar Disorder: Without Psychosis

FIGURE 3 | Portal output based on two separate search criteria. (A) Search criteria, **(B)** Results of IBVD for each of the queries displayed in tabular as well as z-score plot form, **(C)** Publication results for the queries, **(D)** Entrez Gene result along with the top five most published genes for the disorder and brain region in query, **(E)** PubBrain results for the disorder queried which enlist the brain regions published for that disorder, **(F)** Data resources which can provide the user with MRI data available for the disorder queried as well as normal control data which fits the rest of the query criteria.

1000 functional connectomes (fCON1000) project and the OASIS longitudinal and cross-sectional datasets do not make it explicitly clear in their demographics the diagnosis of their subjects. This information needs to be inferred based on the description provided on their respective webpages as normal controls for the fCON1000, and as probable Alzheimer's Disease if a CDR scores >0 or otherwise healthy controls for the OASIS datasets. Similarly, the OASIS datasets, which actually are available for download through XNAT central, do not provide the gender of the subjects on XNAT central, hence in case of gender specific query, the XNAT central resource gets ignored. As of this writing, PING and NIH pediatric database are not yet available for direct query over the web. In these cases, we have separate access to the demographic information saved locally upon which those specific queries are run. CANDIShare, fCON, and OASIS databases are available through XNAT (NITRC-IR and XNAT central) and have some similarities between their demographics data structure. However, in general we had to run individualized queries for most of the databases, making it an *ad-hoc* peer-to-peer style process as described earlier. We hope that in future there would be developments toward streamlining and homogenizing the way the information in stared and presented. The INCF Neuroimaging Data Sharing task force (Poline et al., 2012) is working on an API which would standardize description of neuroimaging/meta data to facilitate the communication between databases. However, we are still far away from standardization of available research resources, hence necessitating a portal presented in this paper.

FUTURE WORK

We will continue expanding the list of available resources as and when they become available and open to be queried. fMRI activation results from the BrainMap (Laird et al., 2005) and SuMSDB (Van Essen et al., 2004) databases are an obvious extension. In addition, bridging between resources that integrate across species will be critical. Adding homology mapping and additional resources like the Allen Brain Institute mouse gene expression database and the CoCoMac database of connectivity will broaden the types of inference that can be supported by the portal environment. We plan to further customize our currently reported results, similar to the brain volume z-score plots or the most published genes, to improve the end usability.

When searching for neuroimaging data using the portal, the user quickly runs into the barriers of publically vs. privately shared data sources. While the portal helps to identify the magnitude of query results that will be found if one has access to these private data sources, users themselves must conform to the various data sharing policies needed for each. Future extensions to the portal that help a user manage their multiple different resource access permissions and facilitate data integration across these multiple sites will be pursued.

In the near future we plan to add a feature to highlight the queried aberrance in the Z-plots. In this fashion, it will become clearer where there is or isn't data available and how that age-range-specific data fits in the context of data from other ages.

The portal currently takes into consideration one disorder and one brain region of interest. In future, we plan to add additional number of disorders to address co-morbidities. We also plan to expand the query to include more than one ROI, so that any commonalities in the results that might exist between multiple brain regions, which could shed more light on the etiology of a disorder, can be made available to the user.

CONCLUSION

Despite of these limitations, our portal provides an initial prototype for a homogenized front end for a variety of resources that would ease the burden of information integration for child-psychiatry researchers.

ACKNOWLEDGMENTS

This work was supported by NIH-NIMH R01 MH083320 "A Knowledge Environment for Neuroimaging in Child Psychiatry" PI(s): Frazier/Kennedy and NIH-NICHD P30 HD004147 "Intellectual and Developmental Disabilities Research Center" PI: McIlvane.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/journal/10.3389/fninf.2014.00047/abstract

REFERENCES

Biswal, B. B., Mennes, M., Zuo, X. N., Gohel, S., Kelly, C., Smith, S. M., et al. (2010). Toward discovery science of human brain function. *Proc. Natl. Acad. Sci. U.S.A.* 107, 4734–4739. doi: 10.1073/pnas.0911855107

Brown, T. T., Kuperman, J. M., Chung, Y., Erhart, M., Mccabe, C., Hagler, D. J., et al. (2012). Neuroanatomical assessment of biological maturity. *Curr. Biol.* 22, 1693–1698. doi: 10.1016/j.cub.2012.07.002

Cachat, J., Bandrowski, A., Grethe, J. S., Gupta, A., Astakhov, V., Imam, F., et al. (2012). A survey of the neuroscience resource landscape: perspectives from the neuroscience information framework. *Int. Rev. Neurobiol.* 103, 39–68. doi: 10.1016/B978-0-12-388408-4.00003-4

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.* 74, 829–836.

Cleveland, W. S., and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *J. Am. Stat. Assoc.* 83, 596–610.

Di Martino, A., Yan, C. G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., et al. (2013). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Mol. Psychiatry* 18:78. doi: 10.1038/mp.2013.78

Evans, A. C. (2006). The NIH MRI study of normal brain development. *Neuroimage* 30, 184–202. doi: 10.1016/j.neuroimage.2005.09.068

Fair, D. A., Nigg, J. T., Iyer, S., Bathula, D., Mills, K. L., Dosenbach, N. U., et al. (2012). Distinct neural signatures detected for ADHD subtypes after controlling for micro-movements in resting state functional connectivity MRI data. *Front. Syst. Neurosci.* 6:80. doi: 10.3389/fnsys.2012.00080

Gardner, D., Akil, H., Ascoli, G. A., Bowden, D. M., Bug, W., Donohue, D. E., et al. (2008). The neuroscience information framework: a data and knowledge environment for neuroscience. *Neuroinformatics* 6, 149–160. doi: 10.1007/s12021-008-9024-z

Hall, D., Huerta, M. F., McAuliffe, M. J., and Farber, G. K. (2012). Sharing heterogeneous data: the national database for autism research. *Neuroinformatics* 10, 331–339. doi: 10.1007/s12021-12012-19151-12024

Jack, C. R. Jr., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., et al. (2008). The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods. *J. Magn. Reson. Imaging* 27, 685–691. doi: 10.1002/jmri.21049

Kalar, D., Poldrack, R., Parker, D. S., Torvik, V., Smalheiser, N., and Bilder, R. M. (2007). PubBrain: an interactive website for literature visualization and exploration. *Organ. Hum. Brain Mapp. Abstr.*

Kennedy, D. N., Haselgrove, C., Hodge, S. M., Rane, P. S., Makris, N., and Frazier, J. A. (2012). CANDIShare: a resource for pediatric neuroimaging data. *Neuroinformatics* 10, 319–322. doi: 10.1007/s12021-12011-19133-y

Kennedy, D. N., Haselgrove, C., and McInerney, S. (2003). MRI—based morphometric analysis of typical and atypical brain development. *Ment. Retard. Dev. Disabil. Res. Rev.* 9, 155–160. doi: 10.1002/mrdd.10075

Laird, A. R., Lancaster, J. J., and Fox, P. T. (2005). Brainmap. *Neuroinformatics* 3, 65–77. doi: 10.1385/NI:3:1:065

Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2007). Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J. Cogn. Neurosci.* 19, 1498–1507. doi: 10.1162/jocn.2007.19.9.1498

Poline, J. B., Breeze, J. L., Ghosh, S., Gorgolewski, K., Halchenko, Y. O., Hanke, M., et al. (2012). Data sharing in neuroimaging research. *Front. Neuroinform.* 6:9. doi: 10.3389/fninf.2012.00009

Schwartz, Y., Barbot, A., Thyreau, B., Frouin, V., Varoquaux, G., Siram, A., et al. (2012). PyXNAT: XNAT in Python. *Front. Neuroinform.* 6:12. doi: 10.3389/fninf.2012.00012

Van Essen, D., Dickson, J., Harwell, J., and Hanlon, D. W. (2004). "SumsDB: online access to surface-based representations of cerebral and cerebellar cortex in primates and rodents," in *Human Brain Project Annual Meeting* (Bethesda, MD).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 19 December 2013; accepted: 07 April 2014; published online: 30 April 2014.
Citation: Rane P, Haselgrove C, Hodge SM, Frazier JA and Kennedy DN (2014) Structure-centered portal for child psychiatry research. Front. Neuroinform. ***8****:47. doi: 10.3389/fninf.2014.00047*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 12 June 2014
doi: 10.3389/fninf.2014.00055

NeuronDepot: keeping your colleagues in sync by combining modern cloud storage services, the local file system, and simple web applications

Philipp L. Rautenberg[1,2]*, Ajayrama Kumaraswamy[1], Alvaro Tejero-Cantero[3], Christoph Doblander[4], Mohammad R. Norouzian[4], Kazuki Kai[5], Hans-Arno Jacobsen[4], Hiroyuki Ai[5], Thomas Wachtler[1] and Hidetoshi Ikeno[6]

[1] *Department of Biology II, G-Node, Ludwig-Maximilians-Universität München, Planegg-Martinsried, Germany*
[2] *Department for Innovations, Max Planck Digital Library, München, Germany*
[3] *MRC ANU, Department of Pharmacology, University of Oxford, Oxford, UK*
[4] *Department of Informatics, Technische Universität München, München, Germany*
[5] *Department of Earth System Science, Fukuoka University, Fukuoka, Japan*
[6] *School of Human Science and Environment, University of Hyogo, Hyogo, Japan*

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Daniel Gardner, Weill Cornell Medical College, USA
Kyle H. Ambert, Intel, USA
Gregory A. Book, Hartford Hospital, USA

****Correspondence:***
Philipp L. Rautenberg, Department for Innovations, Max Planck Digital Library, Amalienstraße 33, 80799 München, Germany
e-mail: rautenberg@mpdl.mpg.de

Neuroscience today deals with a "data deluge" derived from the availability of high-throughput sensors of brain structure and brain activity, and increased computational resources for detailed simulations with complex output. We report here (1) a novel approach to data sharing between collaborating scientists that brings together file system tools and cloud technologies, (2) a service implementing this approach, called NeuronDepot, and (3) an example application of the service to a complex use case in the neurosciences. The main drivers for our approach are to facilitate collaborations with a transparent, automated data flow that shields scientists from having to learn new tools or data structuring paradigms. Using NeuronDepot is simple: one-time data assignment from the originator and cloud based syncing—thus making experimental and modeling data available across the collaboration with minimum overhead. Since data sharing is cloud based, our approach opens up the possibility of using new software developments and hardware scalabitliy which are associated with elastic cloud computing. We provide an implementation that relies on existing synchronization services and is usable from all devices via a reactive web interface. We are motivating our solution by solving the practical problems of the GinJang project, a collaboration of three universities across eight time zones with a complex workflow encompassing data from electrophysiological recordings, imaging, morphological reconstructions, and simulations.

Keywords: morphology, electrophysiology, imaging, data management, neuroinformatics, cloud services, research data management

1. INTRODUCTION

Science today deals with a "data deluge" caused by the widespread use of high-throughput sensors in experiments, and the ever more complex simulations afforded by increased computational power (Moore, 1965). Both measured and simulated data need to be stored in raw form, preprocessed, contextualized with metadata, organized to facilitate queries, and then analyzed to produce scientific statements. Ideally, peer-reviewed data should also be available for replication and re-analysis to test new hypotheses as knowledge progresses.

In addition, the need for multi-university collaboration is particularly acute in neuroscience being a multilevel discipline. It tackles questions spanning disparate levels of organization such as genes, neurons, circuits, and behavior with a variety of methods including sequencing, electrophysiology, and computer simulations (Shepherd et al., 1998). Projects with such multi-university collaborations benefit from well organized coordination of the participating specialists (Cummings and Kiesler, 2007).

One challenging aspect of project workflows might concern immediate sharing of highly structured and voluminous data across labs. Tasks of such a project workflow can interdepend: a further step of the local work depends on another operation that is remotely carried out. In this case, scientific workflows allow to optimize and then more efficiently execute scientific processes (Ludäscher et al., 2009). For example, analysis results can motivate further collection of experimental data, whereupon it is clearly of advantage that they are made available once they are produced.

Proposals to alleviate the data management overhead frequently might require scientists to change and diminish their local processing workflow in order to be able to offer distributed access for collaborators to participate in the project. We propose here a novel approach which integrates seamlessly the widespread filesystem-based acquisition, analysis, and publication workflows by leveraging proven cloud synchronization technology. Our implementation of this approach, a service called

NeuronDepot, enables researchers to continue interacting with the scientific project data through the filesystem and at the same time opens up the data for further processes in cloud-based web applications. In this way NeuronDepot exploits the existing substantial investment in development, acquisition, and training in local applications with their mature and rich interfaces and local access to data. We illustrate the approach with a deployment of NeuronDepot tailored to the specific needs of the GinJang project (http://projects.g-node.org/ginjang/), a complex use case that combines data from electrophysiological recordings, imaging, morphological reconstructions, and simulations.

Several initiatives have established databases to make neuromorphological or neurophysiological research data publicly available. NeuroMorpho (http://neuromorpho.org/) is a curated inventory of digitally reconstructed neurons. The goal of the project is to provide dense coverage of available reconstruction data for the neuroscience community (Ascoli et al., 2007). The neurodatabase.org project (http://neurodatabase.org) and the Collaborative Research in Computational Neuroscience (CRCNS) site (http://crcns.org) host electrophysiological data that have been specifically selected by contributing labs for the purpose of making the data available to the public. Typically, data in these databases are from studies that have been published and are provided for use in further investigations after they have served their primary purpose. Only a few projects have been designed to support data sharing in collaborative research.

The CARMEN portal (https://portal.carmen.org.uk/) allows neuroscientists to share data and programs from neurophysiological experiments. Data analysis functions are provided as services that can be applied to the data stored in the system (Austin et al., 2011). Data, metadata, and analysis workflows are accessible via a web interface.

The German Neuroinformatics Node (G-Node) provides a platform for management and sharing of neurophysiological data (http://www.g-node.org/data). Users can upload, organize, and annotate data, and make them accessible to other users or the public. Data annotation follows a flexible schema (Grewe et al., 2011) so that any metadata necessary can be entered. An API provides fine-grained data access through common languages like Python or Matlab, enabling data management and collaborative data sharing directly from the scientists' local data workflow environments (Sobolev et al., 2014a,b).

Recently, the International Neuroinformatics Coordinating Facility (INCF) established the INCF Dataspace (http://incf.org/dataspace), a cloud-based file system to share all kinds of neuroscience data.

One of the first databases integrating results from various fields like morphology, physiology, and immunohistochemistry is the Bombyx Neuron Database for assembling and sharing experimental and analytical data (Kazawa et al., 2008). Its integrative approach inspired also the development of NeuronDepot.

2. SCOPE OF THE NeuronDepot APPROACH

In contrast to some of the infrastructure solutions presented above, NeuronDepot does not focus on a particular field or type of data but leaves the specifics of each data type to the well-established working environments of the participating members. NeuronDepot supports the scientist by providing a service that integrates data flows with the corresponding management and data analysis.

Beyond facilitating collaboration, the development of a database to properly store and backup all the data of the project makes it accessible to further projects. Putting data into structured databases facilitates its reuse and enables replication and verification of analyses.

2.1. THE GinJang PROJECT AND ITS WORKFLOW

NeuronDepot was developed around the German–Japanese collaboration GinJang (http://projects.g-node.org/ginjang/). This project provides a perfect opportunity for use-case-driven development and field-testing of the NeuronDepot infrastructure because (1) it involves three universities with several labs across multiple time zones, (2) it deals with different types of data from neuroanatomy and electrophysiology, and (3) it requires quick synchronization and reliable transfer of large quantities of raw data with complex associated metadata, including both recorded data and simulation results.

The GinJang project studies the processing of auditory signals in the honeybee. Honeybees communicate the direction and distance to food sources with hive-mates by waggle dance (Frisch, 1967). The hive-mates detect and process airborne vibration caused by the bee's wingbeat during the waggle dance, which consists of vibration pulses with a highly specific temporal pattern. Several critical interneurons for processing the airborne vibration have been identified (Ai et al., 2007, 2009; Ai, 2010; Ai and Itoh, 2012; Ai and Hagio, 2013). However, the neural processing of these vibration signals has rarely been studied: types and roles of neurons involved, their circuitry, and their development are largely unknown.

Members of the GinJang project also developed a program (SIGEN, see Minemoto et al., 2009) that is used to automatically extract and segment the morphology of interneurons that are involved in vibration processing. Goal of the GinJang project is to clarify the morphological characteristics of the vibration-processing neurons and their morphological development according to age and experience of the bees.

The workflow of the GinJang project is illustrated in **Figures 1**, **2**. The experimental setup is at Fukuoka University where electrophysiological measurements (**Figure 1A**), electrophysiological analyses (**Figure 1D**), and imaging (**Figure 1B**) are performed. The image stacks are used at the University of Hyogo for neuronal segmentation (**Figure 1C**). The resulting 3D neuronal segmentations are then normalized by registering them to the Honeybee standard brain (HSB; http://www.neurobiologie.fu-berlin.de/BeeBrain/Project.html), which is done at Fukuoka University. Finally, morphological analyses, simulations, and further analyses are done at Ludwig-Maximilians-Universität München (LMU) (**Figure 1E**).

2.1.1. Data acquisition

The vibration-sensitive neurons in the honeybee auditory system are electrophysiologically and anatomically characterized at Fukuoka University. Using sharp electrodes, voltage traces are recorded from interneurons in response to several sensory input

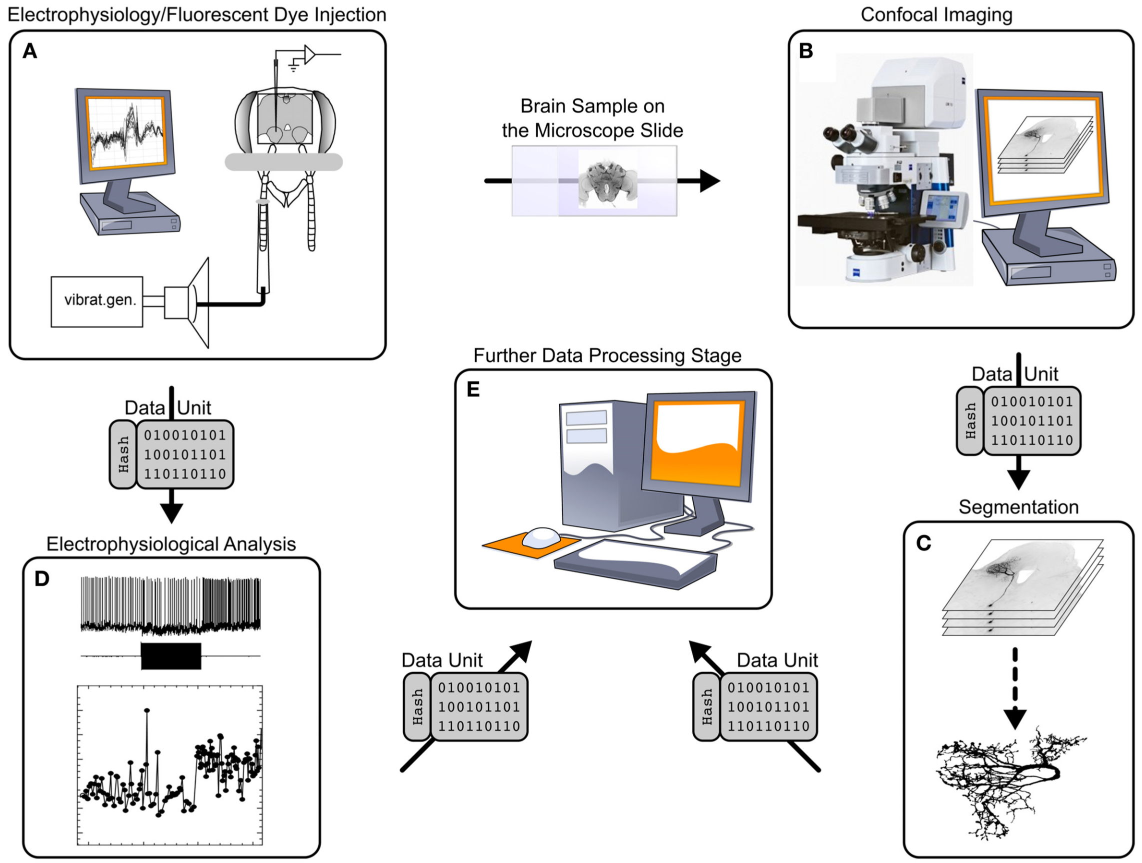


FIGURE 1 | Processing stages and data transitions of a typical workflow like the GinJang honeybee project. (1) Processing stages **(A)** Single cell recording at a electrophysiological setup. Here, the electrical cell activity is measured at the dendrite as well as a dye is injected into the cell. **(B)** Using the brain from experiments, image stacks are created applying confocal microscope technology. **(C)** The application SIGEN computes from confocal image stacks segmentations representing the underlying neuron. **(D)** Electrophysiological recordings are analyzed with specialized software. This stage represents an entire electrophysiological infrastructure using local computers at the experimental lab but also remote G-Node-services. For simplicity, this illustration exemplary shows the result of a spike detection algorithm that identifies spikes of three neuronal units. **(E)** Further process stages follow that build upon already processed data. (2) Traditional data transition (**A** → **B**) The honeybee brain is physically moved from electrophysiological setup to the confocal microscope setup. (**B** → **C**; **C** → **E**) data units (single file, or set of files that represents a logical unit like all files of an image stack) are transferred by common tools like USB-sticks, external hard drives, Dropbox, or simply as email attachments. The same tools are applied for (**A** → **D**; **D** → **E**) but moreover dedicated web techniques for the domain of electrophysiological provided by G-Node can be applied.

protocols (Ai et al., 2009; Ai and Itoh, 2012; Kai et al., 2013). Then the neuron is filled with a dye and imaged at a different setup using confocal microscopy to generate anatomical image stacks (Ai, 2010; Ai and Hagio, 2013). Thus, every experiment generates three kinds of data:

- Electrophysiological data (e.g., voltage and current traces).
- Microscopy image stacks.
- Honeybee metadata (e.g., age or colony) and neuron metadata (e.g., phenotype).

2.1.2. Segmentation

Image stacks are transferred to the University of Hyogo, Japan. Here, using automated image analysis software SIGEN (Yamasaki et al., 2006; Minemoto et al., 2009) the 3D structure of the neuron is extracted and stored using the SWC file format (http://www.neuronland.org/NLMorphologyConverter/MorphologyFormats/SWC/Spec.html). At this stage two kinds of data are generated:

- Segmented neuron (e.g., SWC file).
- Parameters used for segmentation (which constitute segmentation metadata).

2.1.3. Registration

The morphological segmentations of the neurons are transferred back to Fukuoka University for registration into the Honeybee Standard Brain using various transformations. We use the honeybee standard brain to analyze the spatial relationships among morphologically and physiologically characterized vibration-sensitive neurons. The neuronal profiles of stained interneurons, obtained from different preparations, are segmented as explained

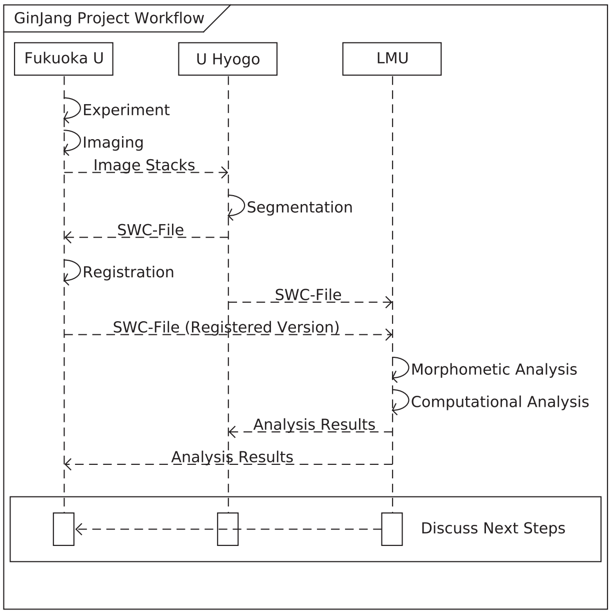


FIGURE 2 | Sequence Diagram of the GinJang workflow (morphological scope). The GinJang workflow starts at the Fukuoka University with two processing stages (indicated by solid arrows): the experimental data collection and the imaging processing stage. Anatomical image stacks are transferred (dashed arrow) to University of Hyogo where they are segmented. Segmented neurons are transferred to the LMU and also back to Fukuoka University where they are registered to the honeybee standard brain. Unregistered segmentation and registered segmentation are used for simulations and analysis at the LMU. Using analysis results, scientists in Fukuoka can tweak existing experiments or design new ones.

in the previous section. Subsequently, the neuropilar outlines are traced semi-automatically with Amira 4.1 (Evers et al., 2005) and ITK-SNAP (http://www.itksnap.org/). These neuropilar label fields are used to register the segmented neuron of each preparation into the honeybee standard brain following the method described by Brandt et al. (2005). Data generated at this stage are:

- Registered neuron morphology.
- Parameters used for registration.

2.1.4. Analysis

These segmentations (both registered and unregistered) are transferred to the LMU, Germany, where 3D segmentations are used for morphometric analysis and simulation studies. Multiple kinds of data are generated at this stage:

- Model files for simulations.
- Simulation metadata, e.g., parameters of simulation, location of stimulation (input) and measurements (output).
- Simulation results: visualizations and summary data.
- Morphometric analysis metadata, e.g., subregion of analysis, metrics used.
- Results of morphometric analysis: volume, surface area, number of branch points.

2.1.5. Traditional data transfer methods

The workflow of the GinJang project requires multiple data transfers between diverse processing stages. These transfers were previously done via e-mail, usb-sticks, external hard drives, ftp-servers, or cloud storage services (like Dropbox, http://www.dropbox.com/). NeuronDepot replaces these traditional data transfer methods.

3. REQUIREMENTS ANALYSIS

We asked the members of the GinJang Project to specify the features that they expect to have in NeuronDepot. Based on those we came up with the following set of requirements:

1. Data Management
   - Replacement of “manual” data transitions that are using memory. devices like e-mail, USB-sticks, external hard drives, FTP-servers, or cloud storage services.
   - Ease of metadata assignment for various kinds of data like image stacks, voltage trace, and neuronal reconstructions.
   - Interrelate various kinds of data.
   - Visibility of the current state of the project through a web browser.
   - Automatic update and synchronization across project workstations.
2. Integration
   - Maintenance of the well-established work environments of the participating scientists.
   - Minimization of the integration effort.
3. Data Security
   - Reliability of upload and download of large data.
   - Access control.
4. Automated Backup
5. Additional Requirements
   - Easy adaptation to new data-specific requirements that emerge during the project.
   - Support for automated data processing like metadata extraction, analysis, and simulation.
   - Quick overview of contents and metadata.
   - Flexible search of data.

4. CONCEPT

4.1. NeuronDepot AS A SERVICE

NeuronDepot is designed as a service. As opposed to a product, functionalities of a service are set up to meet a specific set of requirements at a point in time (Truex et al., 1999; Bennett et al., 2000; Bullinger et al., 2003). Therefore, the specific form of NeuronDepot changes as the project progresses and its requirements continually evolve. Moreover, NeuronDepot brings together other already existing service-modules, which are reassembled and configured to meet the current requirements. While building up NeuronDepot from its sub-services, we make sure that NeuronDepot stays functional even when its sub-services develop with time. Also, these developments can be utilized in evolving NeuronDepot. By offering new functionality in a way that is compatible with existing services, tools, training, and working environments, the costs of data sharing in a collaboration are brought down to a minimum while the accessibility of research assets is future-proofed.

4.2. CORE IDEA

When handling a large amount of data, it is common for scientists to arrange the corresponding files in a directory tree. By doing this, they often encode metadata in the name of directories, for example, the date of recordings or experimental parameters. NeuronDepot also uses this well established principle. The difference is that NeuronDepot automatizes this. It employs a set of rules to automatically create such a directory structure and arrange the data. It uses the associated metadata (available in the database) for naming the directories. The rules for forming this directory structure can be changed. Thus, the same data can be organized in different structures as required by the scientist.

4.3. DEFINITIONS

In this section we define terminologies which are used in explaining NeuronDepot.

4.3.1. Data unit

A data unit (**Figure 3**, bottom left and bottom right) is a logical grouping of one file (trivial case) or multiple files which are generated by a single process. Examples of individual data units in the GinJang context include: an image stack consisting of several image files, the morphology of a neuron represented in a single SWC-file, or several plots and tables resulting from simulations of a neuron's electrophysiological characteristics.

4.3.2. Context path and context trees

Any data unit can be uniquely identified by a subset of the metadata attributes associated with it (**Figure 3**). We define the context path of a data unit as an ordered list of the specific attributes that uniquely identify it (**Figure 4**). This context path can be used to construct a path in the file system where the order of metadata

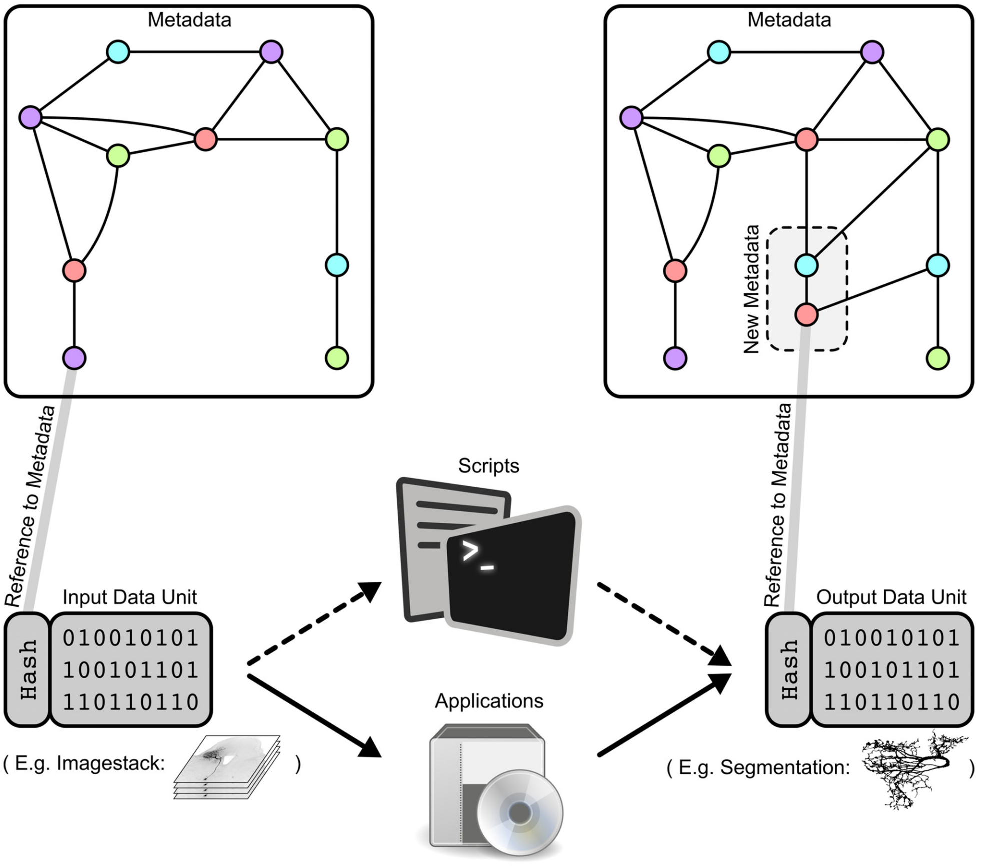


FIGURE 3 | Data units as smallest logical entity for specific data processing attached to the metadata of the project. (Left) A data unit is connected to metadata by its unique hash value id. Metadata are illustrated here as a graph where each point represents an attribute like `AGE=15`, `DATE=130525`, or `HONEYBEE=HB123`. The data unit could express an image stack or a compartmental reconstruction of a specific neuron. (Middle) Data processing by a script or an applications that operates on specific input data units and that generates a new output data unit. For example, SIGEN generates from input data units expressing image stacks neural segmentations as an output data unit containing an SWC-file. (Right) Processing a data unit with a specific script or application leads to an output data unit associated with new metadata that is integrated into the existing metadata graph. According to our SIGEN-example, parameters of the segmentation algorithm are stored within the metadata graph.

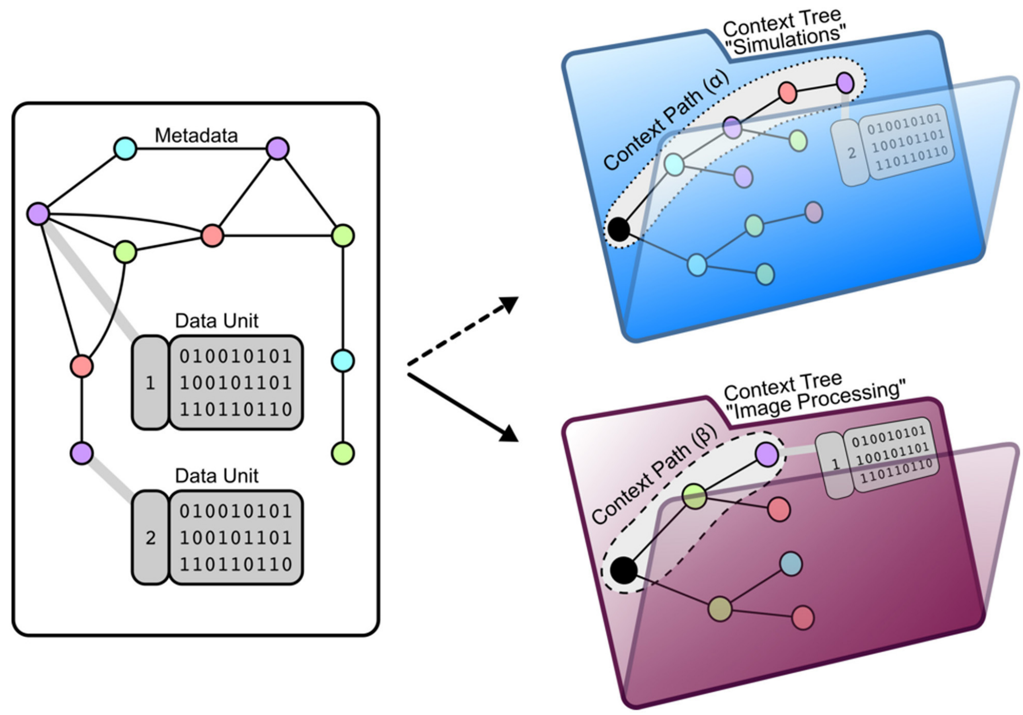


FIGURE 4 | Mapping a data unit to context trees using attached metadata. (Left) Each data unit is connected to metadata. Using this metadata allows the organization of data units by two aspects: The entire data of a project can be (1) sub-divided into divers subsets of data where (2) data units are arranged within a tree structure where the nodes represent metadata and the leafs represent the data unit. As a meaningful specification of such a arrangement depends on the context of data usage, we call this arrangement *context tree*. (Right) One context tree of NeuronDepot arranges project data for the morphometric processing stage. The file format `SWC` serves as a filter argument as just SWC-files are needed for simulation. Metadata attributes `LABOR_STATE`, `REGION`, `HONEYBEE_ID`, and `SIGEN_PARAMETERS` serve for grouping. *Example of context path (α) pointing to a data unit containing a segmentation:* `/forager/left_DL/HB130427/D20V05C01S01/morphology.swc` Another context tree of NeuronDepot arranges project data for our imaging processing stage. Here, the project data are reduced to image stacks. Metadata attributes `HONEYBEE_ID` and `REGION` serve for grouping. *Example of context path (β) pointing to a data unit containing an image stack:* `/HB130427/left-DL/*.tiff`

attributes corresponds to the hierarchical levels in the file system.

Example: If a member of the project wants to analyze one particular segmentation of neuron `NRN-1` of honeybee `HB123`, the following two paths leading to the corresponding data unit would represent these attributes:

```
(1) HB123/NRN-1/segmentation/
(2) segmentation/HB123/NRN-1/
```

The desired order of the attributes depends on how the data units are to be queried for specific analyses: the path order is projected into a hierarchy and therefore defines different grouping levels specific analyses.

4.3.3. Projection

A projection is the representation of a context tree within the file system. It is comparable to materialized views of relational database management systems.

4.4. DESIGN CONSIDERATIONS

The architecture of NeuronDepot follows these principles.

4.4.1. Incorporation of existing open source components

The open source ecosystem holds multiple solutions solving very specific tasks. Some examples are SQLAlchemy[1] (for controlling the persistence of objects by mapping them to database structures), numpy[2] (solving highly optimized numerical tasks), or matplotlib[3] (illustrate data by drawing graphs and figures). Moreover, the community of neuroinformatics has added several domain-specific tools for simulations, analysis, and processing of data from the field. We have incorporated some of these solution while developing NeuronDepot (see section 6). NeuronDepot is also structured so that other such solutions can be integrated to it.

4.4.2. Utilization of established cloud services

Cloud services have rapidly emerged as a widely accepted paradigm built around core concepts such as on-demand computing resources, elastic scaling, elimination of up-front investment, reduction of operational expenses, and establishing a

[1] http://www.sqlalchemy.org/
[2] http://www.numpy.org/
[3] http://matplotlib.org/

pay-per-use business model for information technology and computing services. The use of cloud services helps to reduce development time and effort.

5. HOW NeuronDepot WORKS

5.1. DATA ARRANGEMENT: FLAT ON THE SERVER AND HIERARCHICAL ON USER WORK STATIONS

NeuronDepot applies systematically the principle of using folder names and file names as carrier for metadata describing the data contained in the filesystem. For flexibility, the collection of data units is stored in a central server in a flat structure where each data unit has a unique identifier, and the metadata are kept separate and referenced to those identifiers. When users define a subset of data they are interested in, along with a hierarchical arrangement that suits their needs, NeuronDepot creates a user- and task-specific context tree as a hierarchy of symbolic links with the data units at the leaves. By exposing these hierarchies to a synchronization daemon, the projection is made available to every workstation that subscribes to it.

5.2. DATA ASSIGNMENT

NeuronDepot also leverages advances in synchronization technology for the data upload process: the user simply places new data units in a designated floating folder (comparable to the Camera Upload folder of Dropbox, see section 5.3). This folder is synchronized to the server. Then, the data units appear as available for metadata assignment via a graphical user interface of NeuronDepot. Once metadata assignment is complete, data units can be projected, as described above, to hierarchies that are adapted to the local users' workflows. Data units are now also accessible to cloud analytic services that directly query the metadata database without demanding a specific projection, as these clients are not constrained by the hierarchical data model of filesystems. NeuronDepot thus maintains consistency all the way from the scientists' local copy of acquired data to the cloud-based analysis platforms.

5.3. CLOUD-BASED DATA FLOW

NeuronDepot's mechanism for data transmission is based on synchronization by GWDG Cloud Share (**Figure 5**). This cloud storage service is used to keep all local computers that are involved in the project updated by the server and, therefore, updated among each other. This core update process is based on synchronization on the file system level. In order to integrate data units into workflows, the system provides two types of base folders: floating folders (**Figure 5**-1) and context tree folders (**Figure 5**-3). Floating folders are provided with read/write permissions for project members. Data within floating folders are not assigned to the metadata structure and, therefore, are in a floating state. Floating folders are part of the data-assignment process (**Figure 5**-2). The second type of folder is the context tree folder with read-only permissions for project members that synchronize projected context trees to the local work environment.

The underlying data transfer workflow replaces traditional transfer methods as described above and consists of three steps: (1) new data units are stored within the floating folder and synchronized to the server. (2) Synchronized data units within the floating folder are assigned to the existing project data via a web application. As NeuronDepot's web GUI uses responsive web design it provides optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices from mobile phones to desktop computer monitors (Marcotte, 2010). The system ensures that all data are correctly related to each other and that all data stay consistent. Project members can plug scripts into this assignment process to automate and facilitate data processing. Moreover, the system provides diverse reports to brief the scientists about the current state, or about recent changes. (3) NeuronDepot distributes data units back to project members. According to the underlying context tree, NeuronDepot synchronizes projected context tree folders by cloud storage services to the workstations of the scientists.

6. SYSTEM ARCHITECTURE

6.1. GRAPHICAL USER INTERFACE

The architecture underlying NeuronDepot consists of individual layers and components (**Figure 6**). Users can access NeuronDepot via a web application or through cloud storage synchronization clients. NeuronDepot distinguishes two kinds of users: registered project members which can manage the entire project data and administrators with global permissions including user management. NeuronDepot uses OpenIDs (http://openid.net/) for authentication.

The graphical user interface (GUI) consists of two parts: a web application (**Figure 6A**, left) and local applications (**Figure 6**, right). The web application provides forms for assigning data, entering metadata, annotating data with metadata, and deleting data units and metadata. We used the micro web framework Python-Flask (http://flask.pocoo.org/) for rapid development.

Upload and download processes are handled by GWDG Cloud Share (https://powerfolder.gwdg.de/) incorporated by the virtual filesystem projection layer (see below). Thus, scientists can use established tools like Windows Explorer, Mac Finder, Linux Nautilus, or other file managers to copy files for upload in dedicated folders (**Figure 6A**, right).

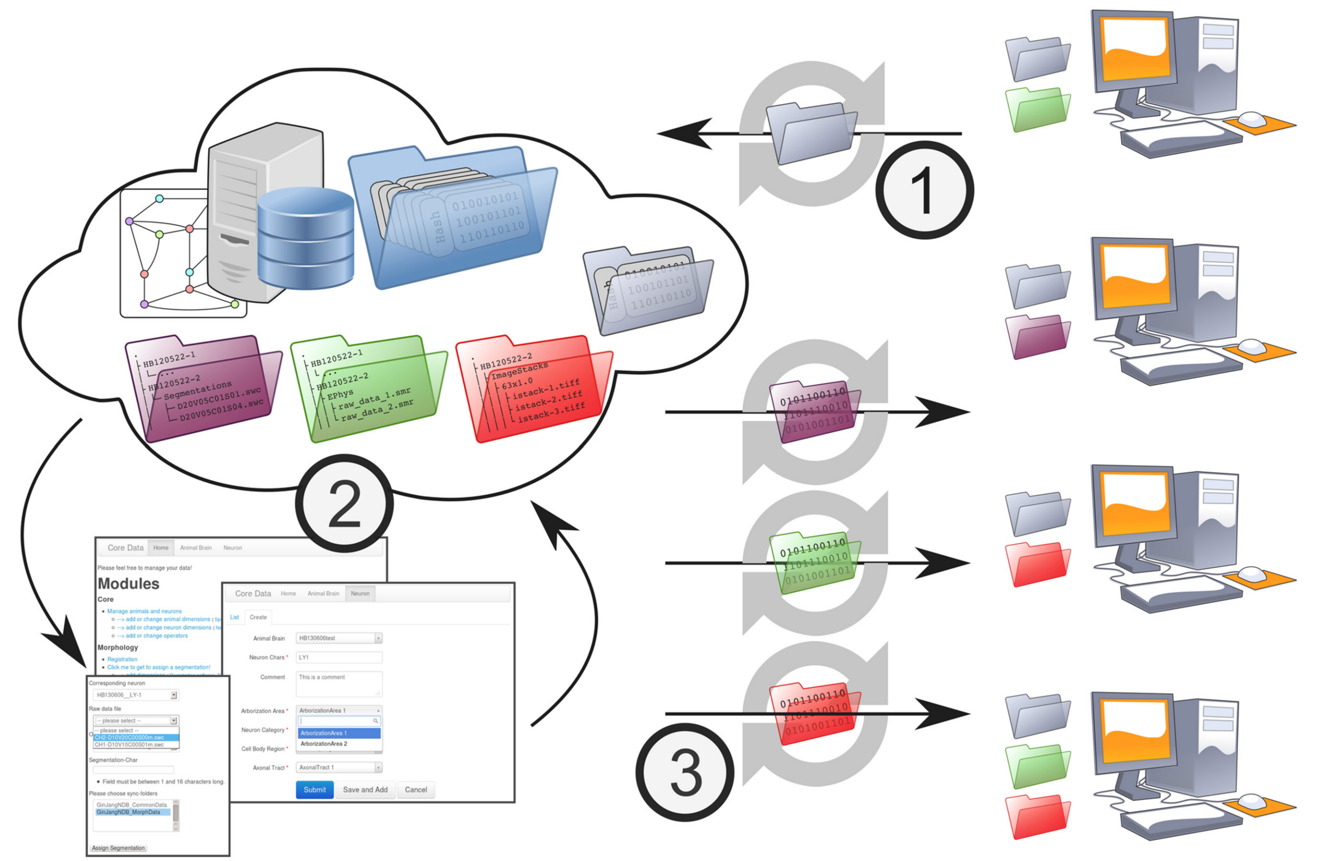
6.2. BUSINESS LOGIC AND VIRTUAL FILESYSTEM PROJECTION

The business logic (**Figure 6B**, left) encodes the NeuronDepot logic rules that determine how data can be created, read, updated, and deleted. We therefore propose a Virtual Filesystem Projection layer (**Figure 6B**, right) which can map data items to cloud storage synchronization clients based on project-specific metadata and provides a consistent view of the file system structure for computational workflows.

A workflow is composed out of multiple tasks. Typically tasks extract metadata, index data items, manipulate images or calculate statistics. Workflows are created in Python with the help of libraries like Snakemake (Köster and Rahmann, 2012).

Workflows are triggered explicitly by user interactions over the web frontend or implicitly by the Virtual Filesystem Projection layer when new files are added. The execution state of the workflow is displayed in the web application.

The filesystem projection layer projects the data items based on metadata to directories and files. The hierarchy of directory tree



Core Data | Home | Animal Brain | Neuron

Please feel free to manage your data!

Modules

Core
- Manage animals and neurons
  - --> add or change animal dimensions
  - --> add or change neuron dimensions
  - --> add or change operators

Morphology
- Registration
- Click me to get to assign a segmentation!

Corresponding neuron: HB130606__LY-1

Raw data file: -- please select --
-- please select --
CH2-D10V20C00S00m.swc
CH1-D10V10C00S01m.swc

Segmentation-Char
- Field must be between 1 and 16 characters long.

Please choose sync-folders
GinJangNDB_CommonData
GinJangNDB_MorphData

Assign Segmentation

Core Data | Home | Animal Brain | Neuron

List | Create

Animal Brain: HB130606test
Neuron Chars *: LY1
Comment: This is a comment
Arborization Area *: ArborizationArea 1
ArborizationArea 1
ArborizationArea 2
Neuron Category *
Cell Body Region *
Axonal Tract *: AxonalTract 1

Submit | Save and Add | Cancel

FIGURE 5 | NeuronDepot and its data flow. NeuronDepot is based on GWDG Cloud Share and simple Flask web-apps that use modern database management systems. GWDG Cloud Share keeps all local computers that are involved in the project synchronized with the server and, therefore, synchronized among each other. This core synchronization is based on the file system. In order to integrate files into workflows, the system provides two types of base folders: floating folders for upload (gray) with read/write permissions for project members and multiple data folders (purple, green, red) with read-only permissions for project members. The data transition workflow consists of three steps: (1) new data units are stored within the floating folder and synchronized to the server. (2) Synchronized data units within the floating folder are assigned to the existing project data via a web application. As NeuronDepot's web GUI uses responsive web design it provides optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices from mobile phones to desktop computer monitors (Marcotte, 2010). The system ensures that all data is correctly related to each other and that all data stay consistent. Project members can plug scripts into this assignment process to automate and facilitate data processing. Moreover, the system provides diverse reports to brief the scientists about the current state or about recent changes. (3) NeuronDepot distributes data units back to project members. According to the underlying context tree, NeuronDepot synchronizes projected context tree folders by cloud storage services to the workstations of the scientists.

is generated by rules using project metadata and the data items which are controlled by the persistence layer.

6.3. PERSISTENCE LAYER

The persistence layer (**Figure 6C**) consists of the following two components:

Project Metadata Within a project database additional metadata are stored. This can be metadata which were extracted by a computational workflow or manually-entered data. For mapping Python objects to database objects we use SQLAlchemy (http://www.sqlalchemy.org/) storing metadata in an PostgreSQL (http://www.postgresql.org/).

Storage Backend The responsibility of the storage backend is to consistently store data items and provide abstractions for the file system projection layer. NeuronDepot uses Camlistore (https://camlistore.org/), which stores files like a traditional filesystem. Moreover, it's specialized in storing higher-level objects.

7. GinJang USING NeuronDepot

In the context of the GinJang project, NeuronDepot manages image stacks and morphological reconstructions of neurons (see section 2.1). The database contains all the image stacks and neuronal reconstructions currently being analyzed as part of the project. It also contains the associated metadata (see section 2.1). The web application presents all the data annotated with metadata in easily readable tables so that the scientists can keep track of it. Such a central presentation of all the data and metadata of the project is useful during the web-based discussions of collaborators in tracking the progress of the project.

By using NeuronDepot, the process of sharing data between the collaborators has been made simple. The data are uploaded at

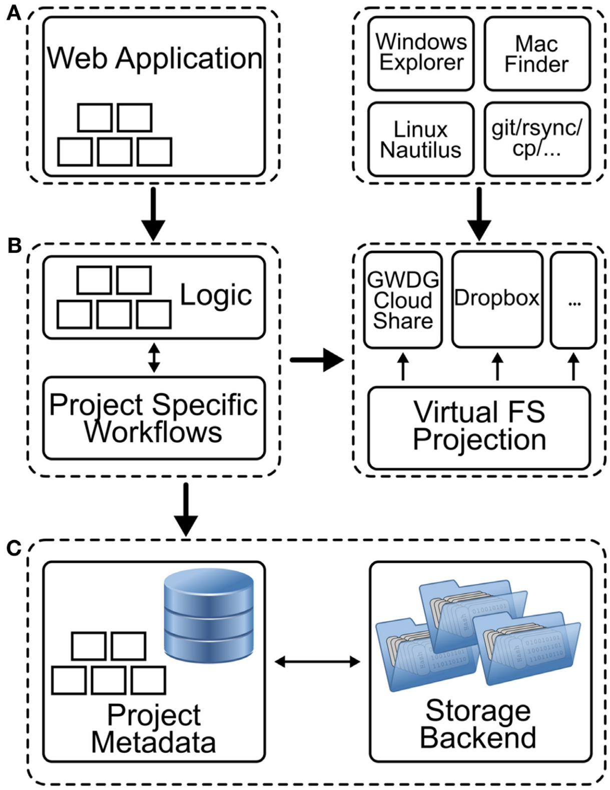


FIGURE 6 | Schematic representation of the different components of NeuronDepot. (A) The graphical user interface consists of two parts: a web application and local applications. The web application (*left*) provides forms for assigning data, enter metadata, and annotating data with metadata. The upload and download processes are handled by GWDG Cloud Share incorporated by the filesystem projection layer (see below). Here, scientists can use established tools like Windows Explorer, Mac Finder, Linux Nautilus, or other file managers to copy files for upload in dedicated folders (*right*) which are connected to the cloud services. **(B)** The business logic (*left*) encodes the NeuronDepot logic rules that determine how data can be created, red, updated, and deleted. Moreover, when new data items are added, deleted, or modified, project-specific workflows can be triggered for each processing stage (*illustrated by five small rectangles*). A Virtual Filesystem Projection layer (right) maps data items and directories to GWDG Cloud Share synchronization clients based on project-specific metadata and provides a consistent view of the file system structure. **(C)** Within a project database additional metadata are stored. This can be metadata which was extracted by a computational workflow or manually entered data. The responsibility of the storage backend is to consistently store data items and provide abstractions for the file system projection layer.

the source of generation once and is automatically made available to the workstations where it is analyzed. Any further changes to this data, for example, if an improved neuronal reconstruction is generated, is automatically made available to the collaborator who is analyzing reconstructions. Thus, data sharing is achieved with minimum manual intervention.

The data assigned to NeuronDepot are analyzed by two collaborators (at University of Hyogo, Japan, and at LMU, Germany), each requiring them in a different hierarchical structure for their analyses. NeuronDepot automatically provides the data in the structure the collaborators specify and thus alleviates the need for manual organization.

8. DISCUSSION

8.1. ADAPTABILITY

The system architecture of NeuronDepot can be conceptually divided into two parts: the core engine, which is not specific to any processing stage, and plain and focused modules, which are project-specific. In the GinJang project, segmentation is a processing stage that is implemented as a dedicated plain module storing, analyzing, and reporting segmentation data. Moreover, such a module provides all the required features for the data and metadata produced by this processing stage like connecting it to other existing data in NeuronDepot, handling upload of this data and specifying the information necessary while presenting it to the user.

NeuronDepot can be adapted to other projects by incorporating project-specific plain modules upon its core engine. These plain modules correspond to the different processing stages of a project, while the core engine remains the same.

8.2. DISTINGUISHING FEATURES OF NeuronDepot

NeuronDepot provides data via file system. This opens up a plethora of tools that are available at the local work bench like (1) desktop search using diverse indexing methods (spotlight, locate, Copernic, Google Desktop), (2) file system explorers (for searching and sorting), (3) Backup, (4) Version-Control, (5) Unix-world applications like `grep, find, and tree` (since "everything is a file"), (6) transmission protocols like `ftp, ssh, and http`, and (7) file synchronization services.

An important feature of NeuronDepot is the isolation of the upload process from the GUI. In the conventional upload process the user indicates the file to be uploaded and waits until the upload process is finished. This way of uploading can be very inconvenient when uploading large files (several hundreds of MBs). This problem is further compounded when the network connection is not stable. Our approach solves this problem by isolating the upload process from the data assignment process. The upload process of NeuronDepot consists of two steps. Data is copied into the GWDG Cloud Share and then assigned from there to the database using the GUI. This upload procedure facilitates assisted assignment of data since the data are available beforehand. Certain analysis scripts can be started on the data in the virtual file system and its results can be later used during the assignment of the data via the web-GUI.

At the end user, a subset of the data in the database is presented in a tree structure. Such a representation of a desired subset of the data in a hierarchical structure provides a partitioning/grouping of the data which becomes very handy if the user intends to perform analysis or comparison on a specific subset of the data.

In NeuronDepot, a specific subset of data is encapsulated into an entity via the concept of context trees. Such an encapsulation facilitates management operations in which treatment of the subset of the data as an entity is essential such as referencing, tagging, and sharing. This is very much like a book encapsulating a set of concepts/facts and making them a single entity.

8.3. COMPARISON WITH OTHER SYSTEMS

Other file-based solutions for collaborative data sharing provide access through a web browser like the web platforms of CARMEN (https://portal.carmen.org.uk/) and G-Node (http://www.g-node.org/data). There, manual download is required to access new data when a dataset has been updated, whereas in NeuronDepot the new data are automatically provided locally.

CARMEN enables access to analysis services on its platform (Austin et al., 2011). G-Node provides access to data in a common representation through an API (Sobolev et al., 2014a) with client tools for integration with the scientist's analysis scripts (Sobolev et al., 2014b). NeuronDepot complements these approaches by presenting the data in the usual file system way. This is particularly useful for collaborations between specific labs where all partners know how to access the data.

Unlike with other existing solutions, using NeuronDepot does not require learning a new GUI or any other infrastructure specific usage features since NeuronDepot provides the data as directory trees to the user. Having this feature, project members could keep their established working environments. In other words: NeuronDepot adapts for existing workflows whereas other systems require the scientist to adapt its workflow to the new system.

8.4. FURTHER DIRECTIONS, LIMITATIONS, AND OPEN QUESTIONS

A package/extension for an existing web-framework like Flask or Django can be developed by reorganizing the system components of NeuronDepot. Several existing solutions of the Open Source Ecosystem were used in the development of NeuronDepot and this is a way of contributing back to it. Moreover, it serves as a good building block for the development of new data software.

As explained in section 4.1, NeuronDepot is a service which develops as the associated project progresses. In the context of the GinJang project, extensions to NeuronDepot are being developed which automate morphological analysis and simulations using the neuronal reconstructions.

At the moment, the context trees used to provide the user with data are hard-coded. The user has to communicate with the developers to have different data structures provided. This process can be slow and can prove to be a hindrance to the scientist's work. A service can be incorporated which enables the user to specify what data structure is needed. This would reduce the user's dependence on the developers and also allow the user to quickly adjust the data that are required from NeuronDepot.

9. CONCLUSION

With this software architecture, we contribute an approach to scientific data workflow and specifically a tool to the neuroscientific infrastructure. NeuronDepot's principal merit is that it integrates smoothly with established tools and resolves the transition from local to cloud-based processing. In doing so, it enables researchers to leverage the advantages of cloud services while not requiring them to relinquish control of their data or analysis.

ACKNOWLEDGMENTS

We thank Andrey Sobolev, Adrian Stoewer, and Christian Kellner for discussions and contributions to a previous version. Supported by the Japan Science and Technology Agency (JST) and the Federal Ministry of Education and Research (BMBF, grants 01GQ1116 and 01GQ1302) and a Hyogo Overseas Research Network (HORN) Scholarship.

REFERENCES

Ai, H. (2010). Vibration-processing interneurons in the honeybee brain. *Front. Syst. Neurosci.* 3:19. doi: 10.3389/neuro.06.019.2009

Ai, H., and Hagio, H. (2013). Morphological analysis of the primary center receiving spatial information transferred by the waggle dance of honeybees. *J. Comp. Neurol.* 521, 2570–2584. doi: 10.1002/cne.23299

Ai, H., and Itoh, T. (2012). "The auditory system of the honeybee," in *Honeybee Neurobiology and Behaviors, 2nd Edn.*, eds D. Eisenhardt, C. G. Galizia, and M. Giurfa (Berlin, Heidelberg: Springer Verlag), 269–284.

Ai, H., Nishino, H., and Itoh, T. (2007). Topographic organization of sensory afferents of Johnston's organ in the honeybee brain. *J. Comp. Neurol.* 502, 1030–1046. doi: 10.1002/cne.21341

Ai, H., Rybak, J., Menzel, R., and Itoh, T. (2009). Response characteristics of vibration-sensitive interneurons related to Johnston's organ in the honeybee, *Apis mellifera*. *J. Comp. Neurol.* 515, 145–160. doi: 10.1002/cne.22042

Ascoli, G. A., Donohue, D. E., and Halavi, M. (2007). NeuroMorpho.Org: a central resource for neuronal morphologies. *J. Neurosci.* 27, 9247–9251. doi: 10.1523/JNEUROSCI.2055-07.2007

Austin, J., Jackson, T., Fletcher, M., Jessop, M., Liang, B., Weeks, M., et al. (2011). CARMEN: code analysis, repository and modeling for e-neuroscience. *Proc. Comput. Sci.* 4, 768–777. doi: 10.1016/j.procs.2011.04.081

Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., and Munro, M. (2000). "Service-based software: the future for flexible software," in *Seventh Asia-Pacific Software Engineering Conference (APSEC 2000)*, (Singapore: IEEE Computer Society Press), 214–221. doi: 10.1109/APSEC.2000.896702

Brandt, R., Rohlfing, T., Rybak, J., Krofczik, S., Maye, A., Westerhoff, M., et al. (2005). Three-dimensional average-shape atlas of the honeybee brain and its applications. *J. Comp. Neurol.* 492, 1–19. doi: 10.1002/cne.20644

Bullinger, H.-J., Fähnrich, K.-P., and Meiren, T. (2003). Service engineering-methodical development of new service products. *Int. J. Prod. Econ.* 85, 275–287. doi: 10.1016/S0925-5273(03)00116-6

Cummings, J. N., and Kiesler, S. (2007). Coordination costs and project outcomes in multi-university collaborations. *Res. Pol.* 36, 1620–1634. doi: 10.1016/j.respol.2007.09.001

Evers, J., Schmitt, S., Sibila, M., and Duch, C. (2005). Progress in functional neuroanatomy: precise automatic geometric reconstruction of neuronal morphology from confocal image stacks. *J. Neurophysiol.* 93, 2331. doi: 10.1152/jn.00761.2004

Frisch, K. (1967). "The tail-wagging dance as a means of communication when food sources are distant," in *The Dance Language and Orientation of Bees*, (Cambridge, MA: Belknap Press of Harvard University Press), 57–235.

Grewe, J., Wachtler, T., and Benda, J. (2011). A bottom-up approach to data annotation in neurophysiology. *Front. Neuroinform.* 5:16. doi: 10.3389/fninf.2011.00016

Kai, K., Ikeno, H., Haupt, S. S., Rautenberg, P. L., Wachtler, T., and Ai, H. (2013). "Response properties of auditory interneurons in the honeybee brain," in *Bernstein Conference 2013*, Sept. 24–27 (Tuebingen, Germany).

Kazawa, T., Ikeno, H., and Kanzaki, R. (2008). Development and application of a neuroinformatics environment for neuroscience and neuroethology. *Neural Netw.* 21, 1047–1055. doi: 10.1016/j.neunet.2008.05.005

Köster, J., and Rahmann, S. (2012). Snakemake-a scalable bioinformatics workflow engine. *Bioinformatics* 28, 2520–2522. doi: 10.1093/bioinformatics/bts480

Ludäscher, B., Altintas, I., Bowers, S., Cummings, J., Critchlow, T., Deelman, E., et al. (2009). "Scientific process automation and workflow management," in *Scientific Data Management*, eds A. Shoshani, and D. Rotem (London: Chapman & Hall).

Marcotte, E. (2010). Responsive web design. *A List Apart*, 306.

Minemoto, T., Saitoh, A., Ikeno, H., Isokawa, T., Kamiura, N., Matsui, N., et al. (2009). "SIGEN: system for reconstructing three-dimensional structure of insect neurons," in *Proceedings of the Asia Simulation Conference, JSST2009, CDROM* (Shiga), 1–6.

Moore, G. E. (1965). *Cramming More Components Onto Integrated Circuits*. New York, NY: McGraw-Hill.

Shepherd, G. M., Mirsky, J. S., Healy, M. D., Singer, M. S., Skoufos, E., Hines, M. S., et al. (1998). The human brain project: neuroinformatics tools for integrating, searching and modeling multidisciplinary neuroscience data. *Trends Neurosci.* 21, 460–468. doi: 10.1016/S0166-2236(98) 01300-9

Sobolev, A., Stoewer, A., Leonhardt, A. P., Rautenberg, P. L., Kellner, C. J., Garbers, C., et al. (2014a). Integrated platform and API for electrophysiological data. *Front. Neuroinform.* 8:32. doi: 10.3389/fninf.2014. 00032

Sobolev, A., Stoewer, A., Pereira, M., Kellner, C. J., Garbers, C., Rautenberg, P. L., et al. (2014b). Data management routines for reproducible research using the G-Node Python Client library. *Front. Neuroinform.* 8:15. doi: 10.3389/fninf.2014.00015

Truex, D. P., Baskerville, R., and Klein, H. (1999). Growing systems in emergent organizations. *Commun. ACM* 42, 117–123. doi: 10.1145/310930.310984

Yamasaki, T., Isokawa, T., Matsui, N., Ikeno, H., and Kanzaki, R. (2006). Reconstruction andsimulation for three-dimensional morphological structure of insect neurons. *Neurocomputing* 69, 1043–1047. doi: 10.1016/j.neucom.2005.12.042

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 31 December 2013; accepted: 30 April 2014; published online: 12 June 2014.
Citation: Rautenberg PL, Kumaraswamy A, Tejero-Cantero A, Doblander C, Norouzian MR, Kai K, Jacobsen H-A, Ai H, Wachtler T and Ikeno H (2014) NeuronDepot: keeping your colleagues in sync by combining modern cloud storage services, the local file system, and simple web applications. Front. Neuroinform. ***8****:55. doi: 10.3389/fninf.2014.00055*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
INTEGRATIVE NEUROSCIENCE

TECHNOLOGY REPORT ARTICLE
published: 21 May 2014
doi: 10.3389/fninf.2014.00054

CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research

Tarek Sherif [1†], ***Pierre Rioux*** [1†], ***Marc-Etienne Rousseau*** [1†], ***Nicolas Kassis*** [1], ***Natacha Beck*** [1], ***Reza Adalat*** [1], ***Samir Das*** [1], ***Tristan Glatard*** [1,2] ***and Alan C. Evans*** [1]*

[1] *ACElab, McConnell Brain Imaging Centre, Montreal Neurological Institute, McGill University, Montreal, QC, Canada*
[2] *CREATIS, INSERM, Centre National de la Recherche Scientifique, Université de Lyon, Lyon, France*

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Gully A. P. C. Burns, USC Information Sciences Institute, USA
Padraig Gleeson, University College London, UK

****Correspondence:***
Alan C. Evans, ACElab, McConnell Brain Imaging Centre, Montreal Neurological Institute, McGill University, 3801 University Street, Webster 2B #208, Montreal, QC H3A 2B4, Canada
e-mail: alan.evans@mcgill.ca

†These authors have contributed equally to this work.

The Canadian Brain Imaging Research Platform (CBRAIN) is a web-based collaborative research platform developed in response to the challenges raised by data-heavy, compute-intensive neuroimaging research. CBRAIN offers transparent access to remote data sources, distributed computing sites, and an array of processing and visualization tools within a controlled, secure environment. Its web interface is accessible through any modern browser and uses graphical interface idioms to reduce the technical expertise required to perform large-scale computational analyses. CBRAIN's flexible meta-scheduling has allowed the incorporation of a wide range of heterogeneous computing sites, currently including nine national research High Performance Computing (HPC) centers in Canada, one in Korea, one in Germany, and several local research servers. CBRAIN leverages remote computing cycles and facilitates resource-interoperability in a transparent manner for the end-user. Compared with typical grid solutions available, our architecture was designed to be easily extendable and deployed on existing remote computing sites with no tool modification, administrative intervention, or special software/hardware configuration. As October 2013, CBRAIN serves over 200 users spread across 53 cities in 17 countries. The platform is built as a generic framework that can accept data and analysis tools from any discipline. However, its current focus is primarily on neuroimaging research and studies of neurological diseases such as Autism, Parkinson's and Alzheimer's diseases, Multiple Sclerosis as well as on normal brain structure and development. This technical report presents the CBRAIN Platform, its current deployment and usage and future direction.

Keywords: eScience, distributed computing, meta-scheduler, collaborative platform, interoperability, cloud computing, neuroimaging, visualization

INTRODUCTION

For the past decade, scientists in all fields of research have had to cope with the effects of accelerated data acquisition and accumulation, large increases in study size and required computational power (Bell et al., 2009), and most importantly, the need to connect, collaborate, and share resources with colleagues around the world. This general intensification, often referred to as "Big Data" science, is certainly true in biomedical research fields, such as neuroscience (Markram, 2013; Van Horn and Toga, 2013), and cyberinfrastructure has been proposed as a potential solution (Buetow, 2005). The efforts expended by many research groups in deploying cyberinfrastructures have unquestionably led to the development of successful new research methodologies. Neuroimaging platforms and applications have emerged that address common issues using drastically different approaches; from programmatic frameworks (Gorgolewski et al., 2011; Joshi et al., 2011) to advanced workflow interfaces, abstracting technological decisions away from users to various degrees (Rex et al., 2003; Olabarriaga et al., 2010). These applications excel in addressing different aspects of the problem; workflow building, leveraging data or compute grids, data visualization, collaborative elements (topic reviewed in Dinov et al., 2009). However, as these technologies are often strongly rooted in local requirements, they tend to form application and infrastructure "silos," not easily adaptable to needs other than those for which they were originally conceived. Therefore, while the global nature of current scientific collaborations requires broader integration and platform interoperability, efficient integration of heterogeneous and distributed infrastructures across multiple technological administrative domains, in a sustainable manner, remains a major logistical challenge.

Over the past two decades, the evolution of neuroimaging research has led to the development of a rich array of data processing tools and complete analysis pipelines (exhaustive listing on the online NITRC[1] repository). However, many of these tools remain unintuitive to the average researcher, as they require a solid understanding of advanced computer systems and display drastically differing underlying philosophies, which limits their potential for growth and adoption. They often require familiarity with command line and scripting techniques, long lists

[1] http://www.nitrc.org

of configuration parameters and knowledge of how to properly prepare data for use as input. Manually processing heavier loads requires skills for data transfers and submission of analysis jobs to remote HPC sites in addition to a solid understanding of the scheduling software environment and policies used at each site. Furthermore, properly scaling these operations for large multi-site projects requires skills beyond all but the most technical research teams. Usability issues such as these lead to poor adoption of standards for tools and techniques, sub-optimal usage of resources, and immense amounts of replication and overhead cost. This alone represents a sufficient motivation to promote usage of common tools deployed in shared controlled environments where provenance details of each action are carefully recorded to ensure the reproducibility of results (Mackenzie-Graham et al., 2008).

The CBRAIN platform (http://www.cbrain.mcgill.ca) is a web-based, collaborative research platform designed to address the major issues of Big Data research in a single consistent framework. CBRAIN was conceived at a time when the question was no longer of creating resources such as HPC clusters and data repositories, since they already existed. Rather it was of creating a platform to leverage currently existing resources in a way that would best benefit the research community at large. Our primary objective was to build a user-friendly, extensible, integrated, robust yet lightweight collaborative neuroimaging research platform providing transparent access to the heterogeneous computing and data resources available across Canada and around the world. These key goals carry significant challenges. To address them, CBRAIN was designed with the following guidelines:

- Convenient and secure web access (no software installation required)
- Distributed storage with automated, multipoint data movement, and cataloging
- Transparent access to research tools and computing (HPC)
- Flexibility to adapt to extremely heterogeneous computing and data sites
- Full audit trail (data provenance) and logs across all user actions
- Lightweight core components, low requirements for deployment and operation
- Scalability (no architectural bottlenecks)
- Maintainability and sustainability by a research-based team
- Full ecosystem security and monitoring

The development of this type of integrated platform required addressing the aforementioned problems as they manifest themselves in brain imaging research. For example, pipeline tools are often built with hard-coded interactions to a particular cluster scheduling system, showing little understanding of proper HPC usage or consideration for site-to-site portability. This leads to a massive waste of resources as the generated workloads must be re-encapsulated for responsible use of public or shared HPCs. In addition, procedures and policies at various HPC sites, even within the same organization, can differ significantly, imposing additional burden on users and platform builders. Although sites may claim to use the same scheduling software, different scheduling policies may be implemented; queue limits and priorities vary, installed libraries and environment configuration vary, location and performance of various local storage may differ greatly.

In order to foster more flexible national and international collaborations, we seek to extend CBRAIN past these technological borders. CBRAIN was built in several layers, with a focus on ensuring tight coordination of the entire ecosystem: abstraction of extremely heterogeneous computing resources scattered over large distances; abstraction of remote data resources and a collaborative portal entirely accessible from a regular web browser where users can securely control and share, as desired, data, tools, and computing resources. In this paper, we will discuss how the above philosophy and guidelines have been implemented in CBRAIN and we will present the current deployment and usage of the platform within our neuroimaging community.

MATERIALS AND METHODS

CBRAIN OVERVIEW

CBRAIN is a multi-tiered platform composed of three main layers (see **Figure 1**): (i) the access layer, accessible through a standard web-browser (for users) or a RESTful Web API (for applications or other platforms), (ii) the service layer which provides portal services for the access layer, the metadata database, which stores information about all users, permissions, and resources, and orchestration services for resource coordination (users requests, data movement, computing loads, jobs, data models,...), and finally (iii) an infrastructure layer consisting of networked data repositories and computing resources. An arbitrary number of concurrent data sources (data providers), computing sites (execution servers), and CBRAIN portals may co-exist, with only the metadata database as a central element for a given deployment. A data-grid mechanism with synchronization status tracking has been designed to avoid transfer bottlenecks and ensure scalability. Data transfers are coordinated directly from data providers by execution servers, ensuring that data are not transferred through the central service orchestration layer during operation, and that remote data providers are not overwhelmed by direct connections from processing nodes. Data visualization, being handled directly by a CBRAIN portal server, is the only major service that requires a data transfer to the central servers. This core flexibility allows a wide array of possible site setups. The simplest being the creation of a Virtual Site (also referred to as Virtual Organization or VO) and associated user accounts. These users will obtain access to CBRAIN shared storage and computing resources, but their data will remain private unless they explicitly decide otherwise. Sites can also integrate their own data providers and/or computing resources (again, shared or private). In addition to hosting private data providers and computing servers, a site may host its own CBRAIN portal within the walls of its institution and explicitly limit all operations to private local resources and private network. Such a configuration ensures a completely local handling of scientific data while at the same time benefiting from the advantages of the platform.

The CBRAIN web portal allows users to authenticate and manage their data and analyses. It also provides several advanced visualization tools for exploring results and performing quality

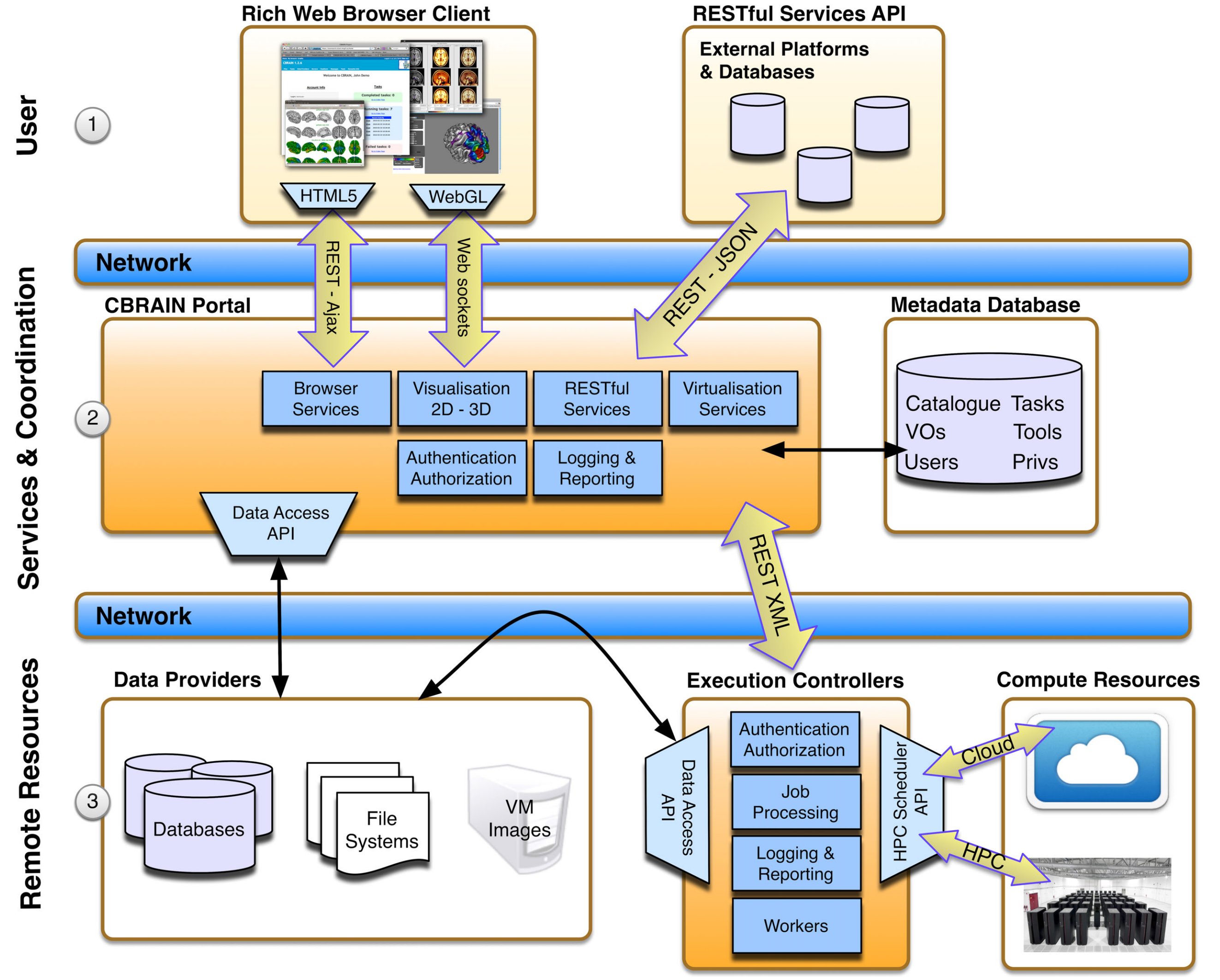
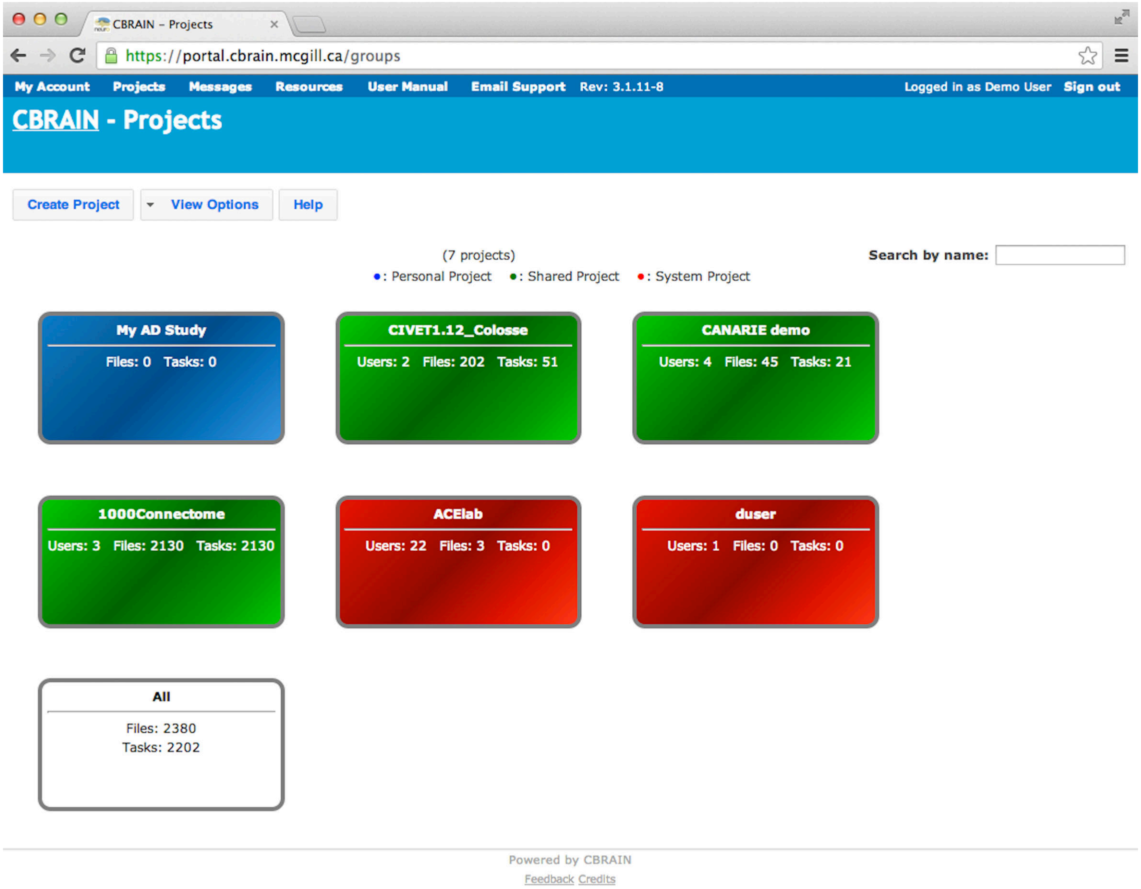


FIGURE 1 | CBRAIN architectural layers. The top user layer (1) represents consumption of services through web browser clients or RESTful API. The central services and coordination layer (2) hosts CBRAIN portals that are responsible for providing services and business logic for requests from the top user layer and orchestration for the lower resource layer. The state of all model instances (users, VOs, tools, resources, catalog, privileges, etc.) is stored in the metadata database. In the lower remote resource layer (3) lays the data providers (scientific data servers, databases or virtual machine images, and tools repositories) and the execution controllers. Execution controllers have to be located at the computing sites on a node that has access to the system scheduler and cluster file systems. Note that data transfers between data providers and execution controllers are triggered by the coordination layer, but do not pass through this layer.

control. The main components of the user environment are shown in **Figures 2–4**; namely the project view, file view and task view. Data is organized in user-created personal or shared projects (**Figure 2**). The file view (**Figure 3**) shows all data files and associated results registered in a selected project from all physical storage locations. Once files or collections are registered in the platform, users can filter, manage, tag, move, and share them across physical locations through a graphical user interface and without having to manage authentication, hostnames, and paths. The same principle applies to tool usage; the user simply selects a set of files and a tool, fill a tool parameter form and launches jobs to be executed remotely. All data transfers, environment setup, scheduler interactions, and monitoring are handled behind the scenes by CBRAIN. Current tasks (sets of computing jobs from various user operations) can be monitored, managed, and troubleshot, if desired, from the task view (**Figure 4**). Once completed, output files appear in the file view as children of the input files (see **Figure 3**). Complete audit trails (provenance) are available for all user actions: logins, file movement and transfers, task parameters, tool versions and logs, resources used, work directories. Links between input files (parents), compute jobs and output files (children) are maintained to allow convenient event browsing when doing post-analysis investigation. Resource views show the status of all data and computing resources accessible to the user (**Figures 5, 6**). The portal also provides a RESTful Web API that exposes CBRAIN functionality to other systems (**Figure 1**). This



(7 projects)

●: Personal Project ●: Shared Project ●: System Project

My AD Study
Files: 0 Tasks: 0

CIVET1.12_Colosse
Users: 2 Files: 202 Tasks: 51

CANARIE demo
Users: 4 Files: 45 Tasks: 21

1000Connectome
Users: 3 Files: 2130 Tasks: 2130

ACElab
Users: 22 Files: 3 Tasks: 0

duser
Users: 1 Files: 0 Tasks: 0

All
Files: 2380
Tasks: 2202

FIGURE 2 | CBRAIN portal: project view. Authenticated users can see a representation of the various projects they own. Projects are color coded: blue for personal projects, green for shared projects, red are default user or site projects, and white allows access to all files owned by this user.

API allows decoupled cross-platform interoperability; any authorized system may authenticate, exchange data, and launch jobs on CBRAIN.

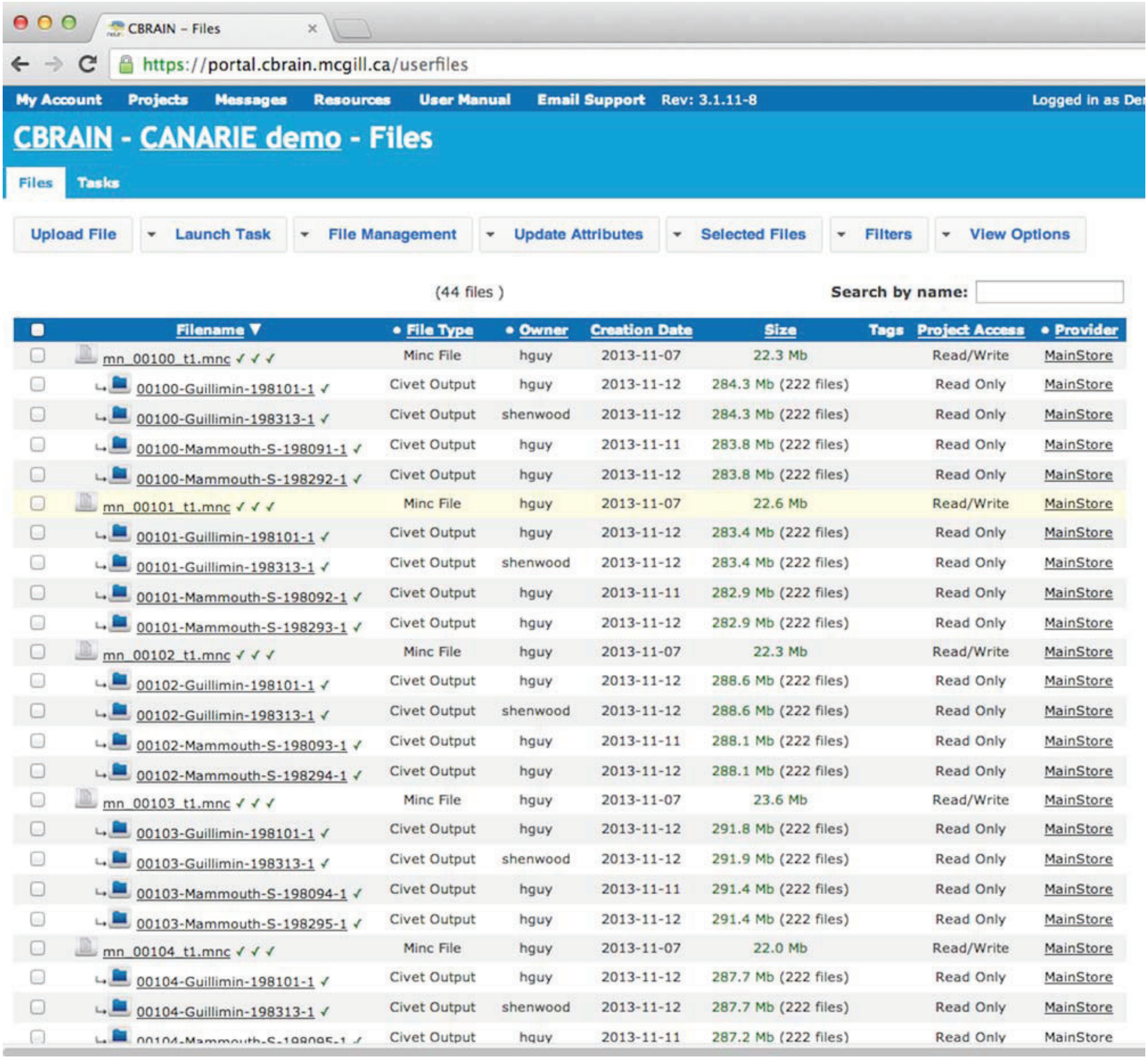
Access policies that regulate the use of CBRAIN-mediated resources for any given project are beyond the scope of this report since access restrictions do not arise from technical limitations. CBRAIN provides flexible capabilities to enforce data access and transfer policies on any computing resource, data source or tool, limiting access to specific users or groups and preventing actual scientific data or services to cross specific boundaries (such as institutional networks) whenever required.

DISTRIBUTED COMPUTING

Computing servers or HPCs connected to CBRAIN run a lightweight execution server. The execution server awaits requests for job submission, performs any setup required by the HPC site and then forwards the job submission request and parameters to the HPC's scheduler. The first challenge faced by CBRAIN was to manage the heterogeneity of these compute resources. Frequently, computing sites are built independently using different architectures, cluster job scheduler software, UNIX environments, storage setups, and overarching usage policies. Developing a centralized point of access that would be reasonably easy to use meant these differences in system architecture had to be overcome in a way that is invisible to the user. CBRAIN addresses this problem in several abstraction layers. The first layer is the Simple Cluster Interface in Ruby (SCIR), a custom library developed in-house.

SCIR was developed as a streamlined meta-scheduler to abstract scheduler differences away from the core platform. SCIR is a simple Ruby library that implements basic high-level functionality required to query, submit, and manage jobs to a given cluster job scheduler. It is implemented with a plugin architecture that makes it easily adaptable to new environments. New grid environments are supported by creating simple SCIR subclasses in Ruby implementing the base SCIR API. SCIR subclasses currently implemented provide support for current and legacy versions of SGE, PBS, Torque, MOAB, and several custom managers and direct UNIX environments.

CBRAIN execution servers simply run in a regular user account on a cluster head node. An execution server on a given HPC receives requests from the CBRAIN portal containing information about the requesting user, the location of data required for analysis, tools and parameters to use, and the data provider



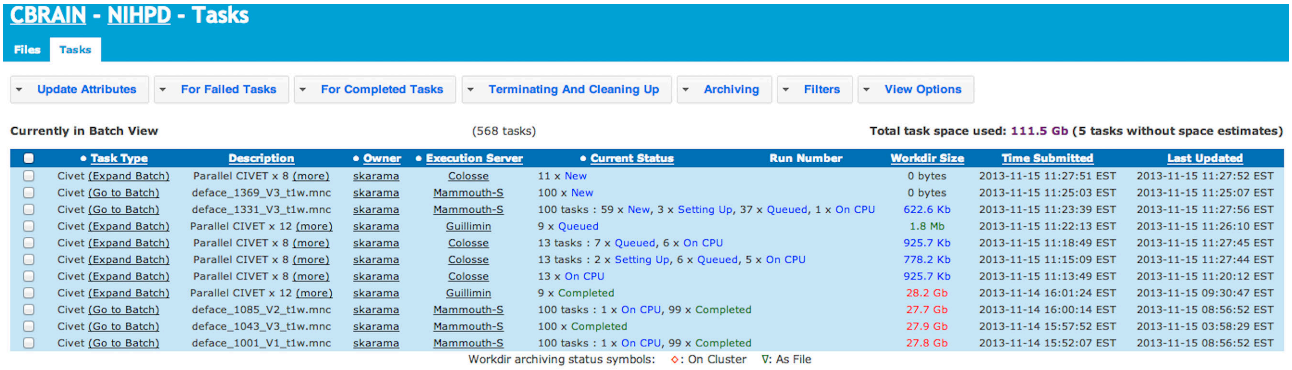
☐	Filename ▼	• File Type	• Owner	Creation Date	Size	Tags	Project Access	• Provider
☐	mn_00100_t1.mnc ✓ ✓ ✓	Minc File	hguy	2013-11-07	22.3 Mb		Read/Write	MainStore
☐	↳ 00100-Guillimin-198101-1 ✓	Civet Output	hguy	2013-11-12	284.3 Mb (222 files)		Read Only	MainStore
☐	↳ 00100-Guillimin-198313-1 ✓	Civet Output	shenwood	2013-11-12	284.3 Mb (222 files)		Read Only	MainStore
☐	↳ 00100-Mammouth-S-198091-1 ✓	Civet Output	hguy	2013-11-11	283.8 Mb (222 files)		Read Only	MainStore
☐	↳ 00100-Mammouth-S-198292-1 ✓	Civet Output	hguy	2013-11-12	283.8 Mb (222 files)		Read Only	MainStore
☐	mn_00101_t1.mnc ✓ ✓ ✓	Minc File	hguy	2013-11-07	22.6 Mb		Read/Write	MainStore
☐	↳ 00101-Guillimin-198101-1 ✓	Civet Output	hguy	2013-11-12	283.4 Mb (222 files)		Read Only	MainStore
☐	↳ 00101-Guillimin-198313-1 ✓	Civet Output	shenwood	2013-11-12	283.4 Mb (222 files)		Read Only	MainStore
☐	↳ 00101-Mammouth-S-198092-1 ✓	Civet Output	hguy	2013-11-11	282.9 Mb (222 files)		Read Only	MainStore
☐	↳ 00101-Mammouth-S-198293-1 ✓	Civet Output	hguy	2013-11-12	282.9 Mb (222 files)		Read Only	MainStore
☐	mn_00102_t1.mnc ✓ ✓ ✓	Minc File	hguy	2013-11-07	22.3 Mb		Read/Write	MainStore
☐	↳ 00102-Guillimin-198101-1 ✓	Civet Output	hguy	2013-11-12	288.6 Mb (222 files)		Read Only	MainStore
☐	↳ 00102-Guillimin-198313-1 ✓	Civet Output	shenwood	2013-11-12	288.6 Mb (222 files)		Read Only	MainStore
☐	↳ 00102-Mammouth-S-198093-1 ✓	Civet Output	hguy	2013-11-11	288.1 Mb (222 files)		Read Only	MainStore
☐	↳ 00102-Mammouth-S-198294-1 ✓	Civet Output	hguy	2013-11-12	288.1 Mb (222 files)		Read Only	MainStore
☐	mn_00103_t1.mnc ✓ ✓ ✓	Minc File	hguy	2013-11-07	23.6 Mb		Read/Write	MainStore
☐	↳ 00103-Guillimin-198101-1 ✓	Civet Output	hguy	2013-11-12	291.8 Mb (222 files)		Read Only	MainStore
☐	↳ 00103-Guillimin-198313-1 ✓	Civet Output	shenwood	2013-11-12	291.9 Mb (222 files)		Read Only	MainStore
☐	↳ 00103-Mammouth-S-198094-1 ✓	Civet Output	hguy	2013-11-11	291.4 Mb (222 files)		Read Only	MainStore
☐	↳ 00103-Mammouth-S-198295-1 ✓	Civet Output	hguy	2013-11-12	291.4 Mb (222 files)		Read Only	MainStore
☐	mn_00104_t1.mnc ✓ ✓ ✓	Minc File	hguy	2013-11-07	22.0 Mb		Read/Write	MainStore
☐	↳ 00104-Guillimin-198101-1 ✓	Civet Output	hguy	2013-11-12	287.7 Mb (222 files)		Read Only	MainStore
☐	↳ 00104-Guillimin-198313-1 ✓	Civet Output	shenwood	2013-11-12	287.7 Mb (222 files)		Read Only	MainStore
☐	↳ 00104-Mammouth-S-198095-1 ✓	Civet Output	hguy	2013-11-11	287.2 Mb (222 files)		Read Only	MainStore

FIGURE 3 | CBRAIN portal: file view. The file view is the main control space where users can manage file or file collection properties (name, privileges, project, tags, type, physical location), filter and select input files based on any property and select a tool for a given task. Web uploads and downloads can be performed through this page, although private data providers or SFTP transfers are preferable for large data. Synchronization information of a file or collection over various caches and data providers is indicated by a symbol next to the file name.

on which to store the results. The server can then synchronize the data to the HPC and make any preparation required by the tool or the HPC in order to successfully run the analysis. This can include creating work directories or setting up environment variables. The execution server then uses SCIR to optimize, convert, and submit the job requests to the local cluster scheduler. Once analysis is done, the execution server initiates transfers of the results to the data provider selected by the user. The execution server is configured through an administrative web interface where parameters such as scheduling type (by core or by node), number of cores per node, maximum queue occupancy, libraries and environment paths, and cache and scratch directories can be set. CBRAIN also performs meta-scheduling activities, such as monitoring jobs, performing failure recovery, optimizing, and re-packaging job loads to match different cluster environments and buffering excess jobs in a meta-queue when quotas are exceeded.

DISTRIBUTED STORAGE

The CBRAIN data provider is an abstract model representing a data repository securely available to the platform from the Internet. Similarly to SCIR, the data provider is a programmatic

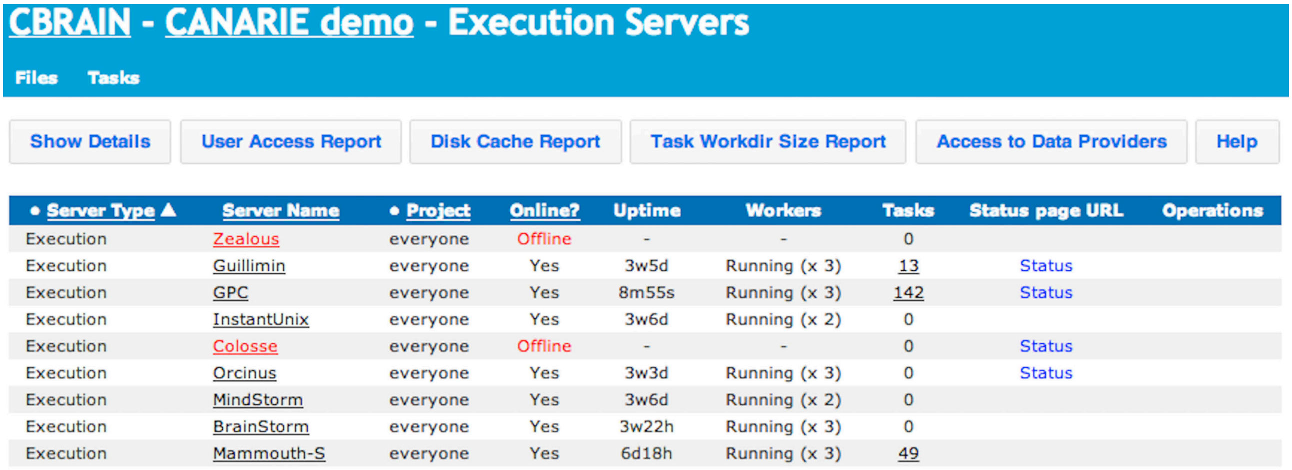


Currently in Batch View (568 tasks) Total task space used: 111.5 Gb (5 tasks without space estimates)

☐	• Task Type	Description	• Owner	• Execution Server	• Current Status	Run Number	Workdir Size	Time Submitted	Last Updated
☐	Civet (Expand Batch)	Parallel CIVET x 8 (more)	skarama	Colosse	11 x New		0 bytes	2013-11-15 11:27:51 EST	2013-11-15 11:27:52 EST
☐	Civet (Go to Batch)	deface_1369_V3_t1w.mnc	skarama	Mammouth-S	100 x New		0 bytes	2013-11-15 11:25:03 EST	2013-11-15 11:25:07 EST
☐	Civet (Go to Batch)	deface_1331_V3_t1w.mnc	skarama	Mammouth-S	100 tasks : 59 x New, 3 x Setting Up, 37 x Queued, 1 x On CPU		622.6 Kb	2013-11-15 11:23:39 EST	2013-11-15 11:27:56 EST
☐	Civet (Expand Batch)	Parallel CIVET x 12 (more)	skarama	Guillimin	9 x Queued		1.8 Mb	2013-11-15 11:22:13 EST	2013-11-15 11:26:10 EST
☐	Civet (Expand Batch)	Parallel CIVET x 8 (more)	skarama	Colosse	13 tasks : 7 x Queued, 6 x On CPU		925.7 Kb	2013-11-15 11:18:49 EST	2013-11-15 11:27:45 EST
☐	Civet (Expand Batch)	Parallel CIVET x 8 (more)	skarama	Colosse	13 tasks : 2 x Setting Up, 6 x Queued, 5 x On CPU		778.2 Kb	2013-11-15 11:15:09 EST	2013-11-15 11:27:44 EST
☐	Civet (Expand Batch)	Parallel CIVET x 8 (more)	skarama	Colosse	13 x On CPU		925.7 Kb	2013-11-15 11:13:49 EST	2013-11-15 11:20:12 EST
☐	Civet (Expand Batch)	Parallel CIVET x 12 (more)	skarama	Guillimin	9 x Completed		28.2 Gb	2013-11-14 16:01:24 EST	2013-11-15 09:30:47 EST
☐	Civet (Go to Batch)	deface_1085_V2_t1w.mnc	skarama	Mammouth-S	100 tasks : 1 x On CPU, 99 x Completed		27.7 Gb	2013-11-14 16:00:14 EST	2013-11-15 08:56:52 EST
☐	Civet (Go to Batch)	deface_1043_V3_t1w.mnc	skarama	Mammouth-S	100 x Completed		27.9 Gb	2013-11-14 15:57:52 EST	2013-11-15 03:58:29 EST
☐	Civet (Go to Batch)	deface_1001_V1_t1w.mnc	skarama	Mammouth-S	100 tasks : 1 x On CPU, 99 x Completed		27.8 Gb	2013-11-14 15:52:07 EST	2013-11-15 08:56:52 EST

Workdir archiving status symbols: ◇: On Cluster ∇: As File

FIGURE 4 | CBRAIN portal: tasks view. The tasks view allows monitoring of task progress, if desired. In this example, the CIVET pipeline has been launched on 1082 MINC files. This workload was split in 568 tasks on 3 different computing sites. CBRAIN has automatically packaged the jobs in proper task units for each execution server. *Colosse* provides full node scheduling with 8 cores per node (Parallel CIVET x8), *Guillimin* has the same type of scheduling, but with 12 cores per node (Parallel CIVET x12), while *Mammouth-S* provides per core scheduling. Although the user has full control of the tasks across the various sites, this is completely optional and transparent. Once jobs are completed, results are automatically transferred to the selected project.



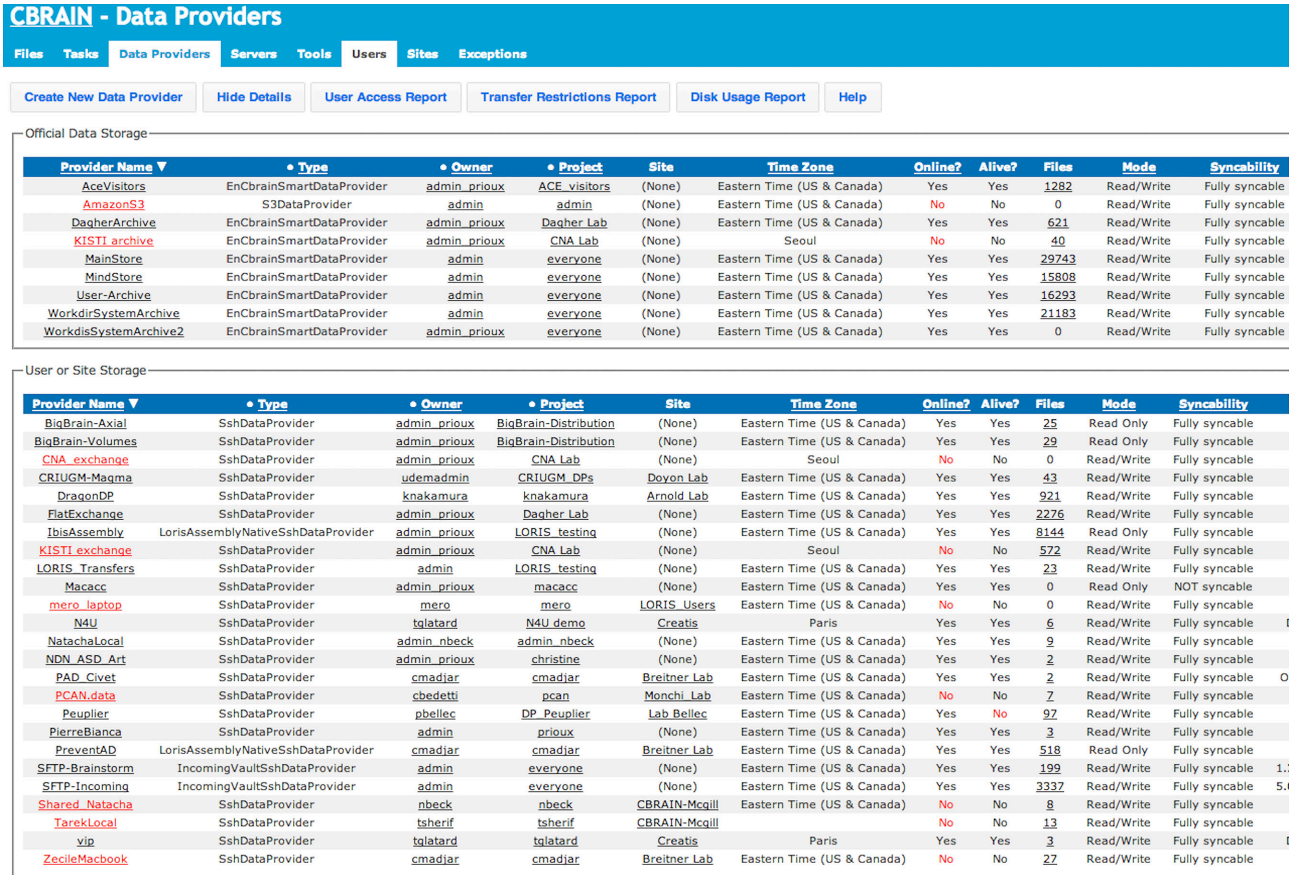
• Server Type ▲	Server Name	• Project	Online?	Uptime	Workers	Tasks	Status page URL	Operations
Execution	Zealous	everyone	Offline	-	-	0		
Execution	Guillimin	everyone	Yes	3w5d	Running (x 3)	13	Status	
Execution	GPC	everyone	Yes	8m55s	Running (x 3)	142	Status	
Execution	InstantUnix	everyone	Yes	3w6d	Running (x 2)	0		
Execution	Colosse	everyone	Offline	-	-	0	Status	
Execution	Orcinus	everyone	Yes	3w3d	Running (x 3)	0	Status	
Execution	MindStorm	everyone	Yes	3w6d	Running (x 2)	0		
Execution	BrainStorm	everyone	Yes	3w22h	Running (x 3)	0		
Execution	Mammouth-S	everyone	Yes	6d18h	Running (x 3)	49		

FIGURE 5 | CBRAIN portal: execution servers view. This view allows users to see which computing resources are available for his/her use and their real-time status. Users can also obtain reports on tool access, cache and data provider utilization, and archived work directories. Administrative users can control group access and put the resource online or offline for CBRAIN users.

API that abstracts away the details of specific types of data stores. The data provider defines a base class of uniform programmatic API methods for querying a file, transferring it, mirroring it and so on, and plugin Ruby classes implement the methods for a particular data store type, allowing CBRAIN to interact with it transparently. CBRAIN widely uses asynchronous data provider wrappers defined for rsync over SSH and SFTP protocols for connecting to data stored remotely. The choice of these tools and protocols does not represent file transfer methodology preferences but rather a pragmatic adoption of the mechanisms commonly supported by data and computing sites. Such mechanisms are also easily manageable by users (site administrators can create a new data provider with the web interface by pointing to the service and adding the CBRAIN public key in the proper account) for greater flexibility and extensibility. These automated grid-like methods have proven robust enough to connect CBRAIN to storage ranging from dedicated network file servers to smartphones. Cloud storage APIs for services such as Amazon S3 and Dropbox, are in the prototyping stage.

A distributed storage model does, however, make network performance a potential concern. CBRAIN makes heavy use of CANARIE's advanced research network[2] and robust synchronization and caching mechanisms were built into the core platform to avoid unnecessary data transfers. The portal and execution servers maintain a local cache of the files that have been asynchronously transferred to them. Synchronization status for all data in all caches is maintained in the metadata database by CBRAIN. Resources will use cached versions of files until the version on the data provider changes, at which point all cached versions will be flagged as invalid. Resources caching invalid data will simply resynchronize with the data provider upon the next requested data operation. Users can manually trigger cache

[2] http://www.canarie.ca



Official Data Storage

Provider Name ▼	• Type	• Owner	• Project	Site	Time Zone	Online?	Alive?	Files	Mode	Syncability
AceVisitors	EnCbrainSmartDataProvider	admin_prioux	ACE_visitors	(None)	Eastern Time (US & Canada)	Yes	Yes	1282	Read/Write	Fully syncable
AmazonS3	S3DataProvider	admin	admin	(None)	Eastern Time (US & Canada)	No	No	0	Read/Write	Fully syncable
DagherArchive	EnCbrainSmartDataProvider	admin_prioux	Dagher Lab	(None)	Eastern Time (US & Canada)	Yes	Yes	621	Read/Write	Fully syncable
KISTI_archive	EnCbrainSmartDataProvider	admin_prioux	CNA Lab	(None)	Seoul	No	No	40	Read/Write	Fully syncable
MainStore	EnCbrainSmartDataProvider	admin	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	29743	Read/Write	Fully syncable
MindStore	EnCbrainSmartDataProvider	admin	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	15808	Read/Write	Fully syncable
User-Archive	EnCbrainSmartDataProvider	admin	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	16293	Read/Write	Fully syncable
WorkdirSystemArchive	EnCbrainSmartDataProvider	admin	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	21183	Read/Write	Fully syncable
WorkdisSystemArchive2	EnCbrainSmartDataProvider	admin_prioux	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	0	Read/Write	Fully syncable

User or Site Storage

Provider Name ▼	• Type	• Owner	• Project	Site	Time Zone	Online?	Alive?	Files	Mode	Syncability
BigBrain-Axial	SshDataProvider	admin_prioux	BigBrain-Distribution	(None)	Eastern Time (US & Canada)	Yes	Yes	25	Read Only	Fully syncable
BigBrain-Volumes	SshDataProvider	admin_prioux	BigBrain-Distribution	(None)	Eastern Time (US & Canada)	Yes	Yes	29	Read Only	Fully syncable
CNA_exchange	SshDataProvider	admin_prioux	CNA Lab	(None)	Seoul	No	No	0	Read/Write	Fully syncable
CRIUGM-Magma	SshDataProvider	udemadmin	CRIUGM_DPs	Doyon Lab	Eastern Time (US & Canada)	Yes	Yes	43	Read/Write	Fully syncable
DragonDP	SshDataProvider	knakamura	knakamura	Arnold Lab	Eastern Time (US & Canada)	Yes	Yes	921	Read/Write	Fully syncable
FlatExchange	SshDataProvider	admin_prioux	Dagher Lab	(None)	Eastern Time (US & Canada)	Yes	Yes	2276	Read/Write	Fully syncable
IbisAssembly	LorisAssemblyNativeSshDataProvider	admin_prioux	LORIS_testing	(None)	Eastern Time (US & Canada)	Yes	Yes	8144	Read Only	Fully syncable
KISTI_exchange	SshDataProvider	admin_prioux	CNA Lab	(None)	Seoul	No	No	572	Read/Write	Fully syncable
LORIS_Transfers	SshDataProvider	admin	LORIS_testing	(None)	Eastern Time (US & Canada)	Yes	Yes	23	Read/Write	Fully syncable
Macacc	SshDataProvider	admin_prioux	macacc	(None)	Eastern Time (US & Canada)	Yes	Yes	0	Read Only	NOT syncable
mero_laptop	SshDataProvider	mero	mero	LORIS_Users	Eastern Time (US & Canada)	No	No	0	Read/Write	Fully syncable
N4U	SshDataProvider	tglatard	N4U demo	Creatis	Paris	Yes	Yes	6	Read/Write	Fully syncable
NatachaLocal	SshDataProvider	admin_nbeck	admin_nbeck	(None)	Eastern Time (US & Canada)	Yes	Yes	9	Read/Write	Fully syncable
NDN_ASD_Art	SshDataProvider	admin_prioux	christine	(None)	Eastern Time (US & Canada)	Yes	Yes	2	Read/Write	Fully syncable
PAD_Civet	SshDataProvider	cmadjar	cmadjar	Breitner Lab	Eastern Time (US & Canada)	Yes	Yes	2	Read/Write	Fully syncable
PCAN.data	SshDataProvider	cbedetti	pcan	Monchi_Lab	Eastern Time (US & Canada)	No	No	7	Read/Write	Fully syncable
Peuplier	SshDataProvider	pbellec	DP_Peuplier	Lab Bellec	Eastern Time (US & Canada)	Yes	No	97	Read/Write	Fully syncable
PierreBianca	SshDataProvider	admin	prioux	(None)	Eastern Time (US & Canada)	Yes	Yes	3	Read/Write	Fully syncable
PreventAD	LorisAssemblyNativeSshDataProvider	cmadjar	cmadjar	Breitner Lab	Eastern Time (US & Canada)	Yes	Yes	518	Read Only	Fully syncable
SFTP-Brainstorm	IncomingVaultSshDataProvider	admin	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	199	Read/Write	Fully syncable
SFTP-Incoming	IncomingVaultSshDataProvider	admin	everyone	(None)	Eastern Time (US & Canada)	Yes	Yes	3337	Read/Write	Fully syncable
Shared_Natacha	SshDataProvider	nbeck	nbeck	CBRAIN-Mcgill	Eastern Time (US & Canada)	No	No	8	Read/Write	Fully syncable
TarekLocal	SshDataProvider	tsherif	tsherif	CBRAIN-Mcgill		No	No	13	Read/Write	Fully syncable
vip	SshDataProvider	tglatard	tglatard	Creatis	Paris	Yes	Yes	3	Read/Write	Fully syncable
ZecileMacbook	SshDataProvider	cmadjar	cmadjar	Breitner Lab	Eastern Time (US & Canada)	No	No	27	Read/Write	Fully syncable

FIGURE 6 | CBRAIN portal: data providers view. This view, presented from an administrative account, shows both the real-time status of official CBRAIN data storages (top) and user created storage (bottom). Main information shown: type of connection, project, owner, time zone, number of registered files and/or file collections, read/write mode, and synchronization mode. Reports for group access, transfer restrictions, and disk usage are available.

deletion of their data if desired. In addition, execution servers use a throttled data transfer model (Park and Humphrey, 2008). To avoid scalability issues, they initiate only a limited number of concurrent data transfer connections.

For most tasks, data stored on a data provider need never be transferred to the central CBRAIN server. Users can keep their data locally, CBRAIN will transfer it directly from their local stores to an HPC cache to run analysis, and then have the results transferred directly back to their data provider. If a lab or an institution has a private HPC with the proper tools connected to CBRAIN, the data need never leave their institution for processing. They can take full advantage of the abstraction provided by CBRAIN while maintaining full control over the location of their data. The only tools that may require that some data be sent to the CBRAIN portal server are the visualization tools as well as browser uploads and downloads for small datasets. To upload or download large datasets, CBRAIN offers SFTP services for users who do not have private data providers.

SECURITY

Users authenticate into the system by first logging into a private account. All communication between clients and the service middleware layer happens over a secure socket layer (SSL). Interactions between the middleware layer and remote resources occur through secure shell tunnels (SSH) with standard 2048 bits key encryption. As many resources used by CBRAIN are outside of our administrative domain, controlling exposure, and potential propagation of intrusions through intermediate machines is a fundamental security concern. CBRAIN uses an on-demand SSH-agent forwarding mechanism to create communication channels between portals, execution servers and data providers, sending all key challenges back to the service layer and closing all channels when not in use. In addition, CBRAIN is equipped with an SSH-agent locking mechanism. Unlocking requests are made by execution servers using a special key stored in the CBRAIN database. Tunnels are thus opened on demand, conditional to the establishment of the proper handshake and closed as soon as the transfer operation is complete. This has several advantages: it eliminates the risks associated with passwords or private keys located on any intermediate machines, it minimizes the duration of open tunnels and it allows platform administrators to carefully monitor whether the key challenges are associated with actual platform operations or possibly suspicious activities.

PERMISSION MODEL

Access to all resources in CBRAIN is managed by three central concepts: users, projects, and sites. Users represent the account of a CBRAIN user. Once authenticated users are granted access to their environment and to any resources to which they have access. By default, users only have access to data they have added through their account. Ownership can be applied to any object within CBRAIN. This can include data, HPC jobs, projects, data providers, and HPCs. Ownership provides full read/write access: a user can rename, move, edit, or unregister any resources they own. On the other hand, if a user is associated to a shared resource through a site or a project, their access to the resource will depend on how it was configured by its owner.

Projects define shared access to resources. All data providers, data, execution servers, and tools in CBRAIN are associated with a project. Projects can have one or more users as members, and members of a given project will have access to the resources associated with it. This system is similar to group permissions in Unix-like operating systems. Users can create and manage their own projects, and by default the resources associated with these projects will be available only to the project's creator. A user can, however, invite other users to their projects, making it possible to share their data, tools, data providers, and execution servers with others.

A CBRAIN site represents a VO such as a laboratory, institution, or a distributed research group that wishes to have some control over how its resources are used in CBRAIN. A site will have users and projects associated with it, and one or more of those users can be given the role of site manager. A site manager has administration capabilities for resources associated with a given site. They can create and manage user accounts, projects, and other resources for their site. Essentially, a site creates an administrative subdomain in CBRAIN over which one or more local site managers can have control.

PLUGINS AND VISUALIZATION TOOLS

Once data has been processed, users often need to visualize their results. This can be for the purposes of performing quality control on a job that was run, or simply to explore the data in a meaningful way. In many processing-centric platforms, this would require a user to transfer large data sets to their computer and run locally installed visualization software. The CBRAIN portal, however, integrates visualization tools that allow users to explore their data in real-time through their web browser, with only the data necessary for the visualization being transferred to the client. At the most basic level, if a data set contains standard images or quality control related text, these can simply be made available for viewing through the browser. More complex visualization tools can be made available through CBRAIN's viewer plugin architecture, which associates file types with viewers. Formats viewable in CBRAIN currently include text, images, video, audio, MINC volumes, MNI 3D objects, and file types supported by Jmol (molecular structures). Display of most supported types involves simply using the appropriate HMTL element. CBRAIN does, however, provide more complex visualizers for MINC volume data and various surface file formats in the integrated BrainBrowser suite of web-enabled visualization tools (demonstration service available at https://brainbrowser.cbrain.mcgill.ca).

The BrainBrowser Surface Viewer (**Figure 7**) is a web-based, real-time 3D surface viewer capable of viewing MNI Object, Wavefront Object, and Freesurfer ASC files. BrainBrowser allows users to view and manipulate 3D surface data in real-time. Color map data can be applied to surfaces, and color thresholds and opacity can be adjusted to ensure proper viewing. The BrainBrowser Surface Viewer is currently being used to provide web access to the MACACC data set (Lerch et al., 2006). The BrainBrowser Volume Viewer (**Figure 7**) is a web-based, slice-by-slice viewer for 3D MINC volumes. The Viewer provides three panels, one each for the sagittal, coronal, and transverse planes. Each panel displays a slice on a given plane at some position in the volume, and the user is allowed to navigate through the volume by moving the cursor within the volume. Four-dimensional fMRI data can be viewed by manipulating time sliders to view the data across time steps. Subjects can be viewed side-by-side and overlaid. Color maps and thresholds can be adjusted to optimize viewing.

TECHNOLOGY USED

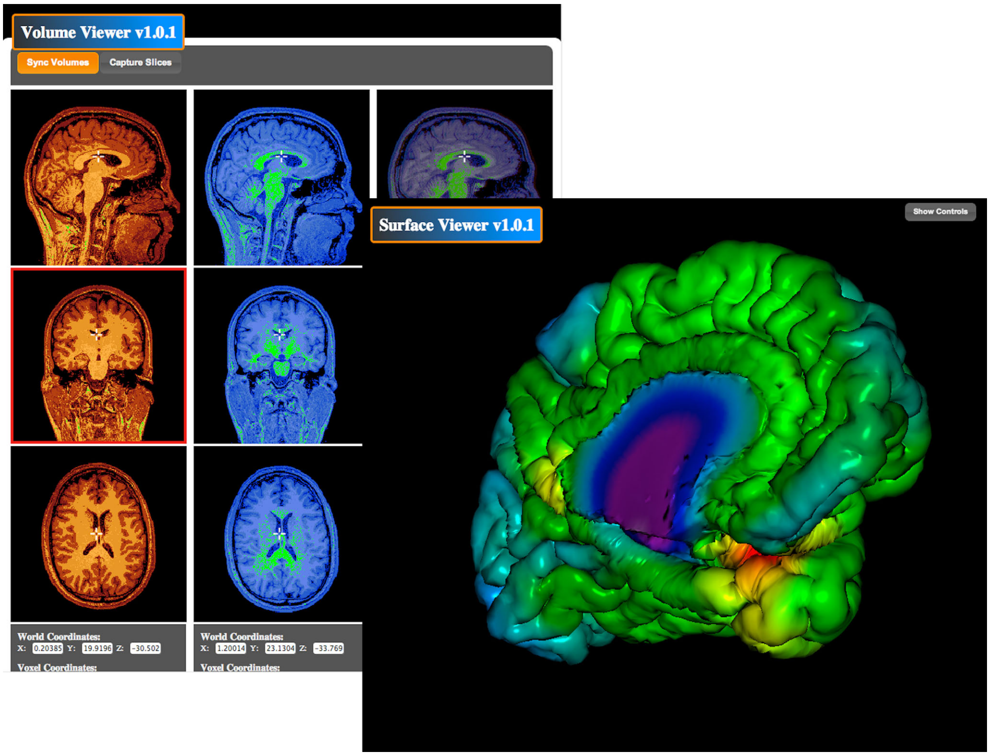
CBRAIN components are implemented using Ruby on Rails[3] (Bachle and Kirchberg, 2007), a widely used RESTful, Ruby-based framework, used by such sites as Github, Twitter, Shopify, Groupon, NASA, Hulu. Our core objective was to follow cutting-edge architecture and development strategies. The key to using Ruby on Rails in a distributed multi-component ecosystem like CBRAIN was streamlining the activities of the various layers and offloading any longer term processing to subsystems. This approach allowed us to take advantage of the built-in object-relational mapping (ActiveRecord) and RESTful nature of Ruby on Rails, while at the same time ensuring that the platform performs and scales elegantly. It also requires less development, hardware, multi-site setups, and operations personnel than common enterprise technologies such as frameworks based on Java. The portal uses Ruby Thin servers behind an Nginx load balancer and a MySQL database to track metadata pertaining to all resources. Using Ruby on Rails also allowed us to develop an agile methodology based on rapid iterations made with constant feedback from users.

CBRAIN development aims to use openly available tools and standards-compliant web technologies whenever possible. This ensures that development and distribution of the system can remain free and unrestricted. All browser interactions with CBRAIN occur over HTTPS and the web client uses standard HTML and CSS for the interface and jQuery[4] and jQuery UI for behavior and theming. The BrainBrowser Volume Viewer uses the HTML canvas element for rendering, and the BrainBrowser Surface Viewer uses three.js[5] for WebGL-based 3D rendering.

[3] http://rubyonrails.org

[4] http://jquery.com

[5] http://threejs.org



World Coordinates:
X: 0.20385 Y: 19.9196 Z: -30.502
Voxel Coordinates:

World Coordinates:
X: 1.20014 Y: 23.1304 Z: -33.769
Voxel Coordinates:

FIGURE 7 | BrainBrowser Surface and Volume Viewers. BrainBrowser allows CBRAIN users to examine any MINC file volume or 3D object (such as surfaces from CIVET, Freesurfer, or Wavefront objects) directly within their web browser. This step enables to conveniently perform quality control, which is often critical before proceeding to further analysis or large data transfers, especially if format conversion steps have been applied.

INTEROPERABILITY

CBRAIN exposes a RESTful Web API to allow interoperability with other platforms and database systems that want to take advantage of its capabilities. Requests are made to the same URLs used for the CBRAIN portal interface using standard HTTP methods (through SSL). The body of an API request can contain XML or JSON and the response will be an XML document representing the data requested. Wrappers for the CBRAIN Web API have been written in Java, Perl, and Ruby. Our usage of Ruby on Rails framework coding conventions ensures that all user interactions with the portal naturally map to RESTful API calls that return XML rather than HTML upon request. This greatly reduces the necessary work required to convert and support the API for cross-platform interoperability.

RESULTS

CURRENT DEPLOYMENT AND USE

CBRAIN has been in active production since 2009 and currently has over 200 users and 80 virtual sites, from 53 cities in 17 countries around the world. Operations are scaled on a yearly basis according to both the yearly computing allocation we obtain and the amount of user support our team can provide. The current production deployment of CBRAIN consists of 12 computing sites, totaling more than 100,000 CPU cores. The infrastructure model is hybrid, while many large clusters are shared national academic research resources (**Table 1**), others sites are institutional or completely private and available solely to CBRAIN. Of these sites, 7 are from the Compute Canada[6] HPC network, 2 are international collaborator sites (Germany and South Korea), and 5 are small local research servers. This integration of heterogeneous resources was done without any new hardware purchases, and does not require administrative access or major changes to local system configuration on the part of the participating sites. Between 2010 and 2013 CBRAIN has launched in excess of 198,000 jobs and obtained an allocation of 13.7 million CPU core hours from Compute Canada alone. CBRAIN provides users with three central data providers, for a total of 80 TB of storage. Furthermore, several user-registered data providers exist as storage for specific projects or institutions. Although it fluctuates significantly, active data currently hosted on the central storage system provided to all CBRAIN users amounts to approximately 13.1 TB in over 100,000 datasets representing 8.4 million files (this does not include computing site caches or user-registered data providers).

CBRAIN provides a wide variety of tools, from pre-processing and analysis pipelines to various file format converters for file types commonly used in neuroimaging research, including MINC, DICOM, NIfTI, and Analyze. Tool integration is prioritized according to the needs of our user community. CBRAIN's philosophy has been to focus on integrating, testing and properly

[6] https://computecanada.ca

Table 1 | CBRAIN high performance clusters and servers.

Machine name	Administrative domain	Location (city, country)	Number of CPU cores
BrainStorm	McGill university	Montreal, Canada	24
Colosse	Compute Canada	Quebec City, Canada	7680
GPC	Compute Canada	Toronto, Canada	30,000
Guillimin	Compute Canada	Montreal, Canada	14,000
Judge	Jülich supercomputing center	Jülich, Germany	2472 + 412 GPU
Juropa	Jülich supercomputing center	Jülich, Germany	17,664
CBRAIN-CNA	KISTI	Seoul, Korea	80
Mammouth-S	Compute Canada	Sherbrooke, Canada	2112
Mammouth-P	Compute Canada	Sherbrooke, Canada	39,648
MindStorm	McGill university	Montreal, Canada	24
Orcinus	Compute Canada	Vancouver, Canada	9600
Zealous	McGill university	Montreal, Canada	24

List of servers and HPCs currently integrated in CBRAIN. All CPUs use standard x86-64 processor architecture and all operating systems are Linux based.

maintaining and supporting tools and features directly requested by our researchers. The platform supports multiple tool versions and the version used in a specific analysis is maintained in the task and provenance logs. Among the most intensively used tools in CBRAIN is CIVET-CLASP (Kim et al., 2005), a processing pipeline for measuring cortical thickness, as well as performing other corticometric and volumetric functions. Components of the popular FSL[7] (Jenkinson et al., 2012), MINC[8], SPM[9], and Freesurfer[10](Reuter et al., 2012) tools have also been integrated. These types of tools are ideal candidates for CBRAIN integration as they are computationally expensive and generally complex to use for the novice user. Most neuroimaging tools have a relatively straightforward workflow, with job inputs and options following a linear sequence of events. However, some pipelines dynamically allocate jobs and dependencies in real-time depending on the inputs they receive. Such job loads have to be carefully analyzed and packaged to ensure optimal use of HPC resources. For example, CBRAIN uses a graph theoretic approach to serialize and parallelize the dynamic job loads of tens of thousands of jobs from NIAK, an fMRI pre-processing pipeline based on the Neuroimaging Analysis Kit for Matlab and Octave, described in Lavoie-Courchesne et al. (2012).

Cross-platform interoperability features have been implemented both in the context of our group's multi-center management system, LORIS (Das et al., 2011) and external collaborative efforts. As part of the "neuGRID 4 you" project (Frisoni et al., 2011), the CBRAIN Web API was consumed by the neuGRID and Virtual Imaging Platform (Glatard et al., 2013) services in Europe using the LONI Pipeline software (Rex et al., 2003). A CBRAIN module for the LONI Distributed Pipeline Server (DPS) was created to interact directly with the CBRAIN Web API. This type of collaboration positions CBRAIN as part of a global network of research platforms, enabling collaborations between users and allowing them to take advantage of the broadest set of services possible.

Although CBRAIN is a generic platform that can accept data and analysis tools from any discipline, its current focus is primarily on structural neuroimaging projects. For example, CBRAIN has been used in a study linking childhood cognitive ability and cortical thickness in old age where DICOM sets from 672 subjects of the Lothian Cohort 1936 were uploaded and registered in CBRAIN from a research group in Scotland, and shared with a group of Canadian researchers for pre-processing and analysis of cortical thickness (Karama et al., 2013). Other examples of initiatives actively using CBRAIN for typical MRI data pre-processing of large cohorts are PreventAD[11], NIHPD[12], NeuroDevNet[13], ABIDE[14], and 1000Brains[15].

DISCUSSION

RELATED WORK

The CBRAIN platform incorporates the key aspects of a grid middleware, namely security (Authentication, Authorization, Accounting—AAA), distributed file management, and job execution on multiple distributed sites. Grid middleware has received a lot of attention in the last 15 years (Foster and Kesselman, 2003), and resulting technologies and concepts are now used in large computing infrastructures such as the Open-Science Grid (Pordes et al., 2007), Teragrid (Catlett, 2002), and the European Grid Infrastructure (Kranzlmüller et al., 2010). CBRAIN is unique in the sense that it integrates all these functions in a single, consistent, lightweight, self-contained, independent framework that is therefore easily administrated and extended. For example, grid security usually relies on X509 certificates signed by trusted authorities, from which time-limited proxy certificates are generated, delegated to the services involved in the platform, and used to authenticate all user operations, for instance job execution and data transfers (Foster et al., 1998). In practice, this mechanism burdens users with the handling of certificates, restricts the range of usable technologies, generates user-specific errors, and complicates debugging. To avoid these issues, CBRAIN decouples user AAA from system AAA: users authenticate to the portal with straightforward login and password, while the portal handles data and computing authorizations, and then authenticates to the services using a single or a few group credentials. Such decoupled approach is being adopted more broadly by portals using so-called robot X509 authentication to infrastructure services (Barbera et al., 2009).

Distributed file management commonly consists of a logical layer providing a uniform view of physical storage distributed over the infrastructure. CBRAIN's file metadata contain similar information to that stored in grid file catalogs, for instance

[7] http://fsl.fmrib.ox.ac.uk/fsl/fslwiki

[8] https://www.nitrc.org/projects/minc

[9] http://www.fil.ion.ucl.ac.uk/spm

[10] http://freesurfer.net

[11] http://www.preventad.com

[12] http://pediatricmri.nih.gov

[13] http://www.neurodevnet.ca

[14] http://fcon_1000.projects.nitrc.org/indi/abide

[15] http://www.fz-juelich.de/inm/inm-1/EN/Forschung/1000_Gehirne_Studie/1000_Gehirne_Studie_node.html

the LCG File Catalog (Baud et al., 2005) or the Globus RLS (Chervenak et al., 2009). However, CBRAIN's file transfer architecture notably differs from the main grid solutions: (i) its throttled data transfer model avoids overloading storage providers, a problem commonly observed in grid infrastructures and addressed in a similar way by the Advanced Resource Connector (Ellert et al., 2007) (ii) it caches files on the computing sites, a feature only provided in a few grid middleware and often implemented at the application level.

Job execution on multiple distributed computing sites is performed either by a meta-scheduler which dispatches jobs to the different sites (Huedo et al., 2001; Andreetto et al., 2008) or by pilot-job approaches provisioning computing resources with generic agents that pull tasks from a central queue when they reach a computing node (Frey et al., 2002; Brook et al., 2003). In neuroimaging, however, due to variations of software and/or libraries, the execution site often has to be controlled by the users to guarantee the correctness and reproducibility of computations (Gronenschild et al., 2012). This is why CBRAIN usually delegates site selection to the users, providing them historical information about queuing times. The matchmaking between tasks and resources, which involves elaborate resource descriptions when performed by a generic grid middleware (Andreetto et al., 2010), is done statically by CBRAIN administrators who map application versions to sites based on their knowledge of the infrastructure.

The decision to develop SCIR as a streamlined meta-scheduler to abstract scheduler differences away from the core platform was based on pragmatic cross-site deployment experience. Libraries with similar goals do exist, but they did not demonstrate enough agility and flexibility for the HPC landscape we faced. The DRMAA (Tröger et al., 2007) and SAGA (Jha et al., 2007) projects, from the Open Grid Forum Working Group, were just emerging standards at the time of the initial CBRAIN deployment. DRMAA is a universal scheduler API library that was used in earlier versions of CBRAIN. Unfortunately, from our experience, although the library defines a fairly complete low-level API, the modules that actually interact with the cluster job schedulers were found to leave certain scheduler versions unsupported and were not designed to be easily extended for interaction with in-house schedulers. Our objectives for low-footprint and flexibility run contrary to dictating scheduler requirements to a diverse array of computing sites, so we created a library suited to our specific needs.

A few other science-gateway frameworks exist to facilitate the building of web portals accessing distributed infrastructures for scientific computing (Marru et al., 2011; Kacsuk et al., 2012). These frameworks provide toolboxes of components meant to be reused in customized assemblies to build domain-specific platforms. To ensure performance and flexibility, CBRAIN developed its own custom portal, which allows fine-grained, optimized interactions with infrastructure services. Other similar leading platforms providing access to neuroimaging applications executed on distributed infrastructures are LONI (Dinov et al., 2010), neuGRID (Redolfi et al., 2009), and A-Brain (Antoniu et al., 2012). While sharing similar overall goals, each platform uses often radically different approaches and philosophy, allowing them to excel in specific niches. For example, LONI offers an advanced and flexible graphical workflow builder that has, to our knowledge, no equivalent in the field. Within CBRAIN, our team took the design decision of supporting only mature, validated workflows as needs arise from our community. CBRAIN users are free to launch any tools or pipelines they have access to, but cannot create and share an automated workflow using multiple tools, the way it would be done in LONI, without contacting the core team. This has the advantage of preventing failures and waste of resources and of enforcing staged validation and quality control, however it does limit the rate of automated workflow integration and flexibility for the users. NeuGRID has a strong remote desktop component capable of providing remote users with native data visualization applications (centralized approach), CBRAIN handles all visualization applications through web-based applications (decentralized approach). These two approaches to the same problem have different characteristics, while the centralized approach procures users with familiar applications in their native mode, supporting usage growth can require large infrastructure investments. The decentralized approach uses very light infrastructure to push modern HTML5 applications to large amounts remote clients, respecting the CBRAIN scalability philosophy, however these applications have to be web compatible or developed anew. The A-Brain platform has done extensive work on low-latency data-intensive processing by building an optimized prototype MapReduce framework for Microsoft's Azure cloud platform on the basis of TomusBlobs (Costan et al., 2013). In comparison, CBRAIN focused on a lightweight, flexible and low-footprint catalog and data grid mechanism that acts as a transparent interface for regular multi-site batch-type projects. While it is clear that the CBRAIN grid cannot move and process multi-terabyte studies with the same ease as A-BRAIN, our goal was to ensure that all user sites can integrate securely in our grid their own data repositories with a minimum of requirements. This leads to a mix of faster and slower storage segments, which CBRAIN manages asynchronously with its caching mechanism. Most of our large imaging projects, with thousands of subjects representing hundreds of gigabytes of data can be processed as-is with the CBRAIN grid. Some multi-terabyte, data-intensive projects, such as our 3D histological reconstruction (Amunts et al., 2013), required special infrastructure for processing and visualization.

The modular plugin approach used to develop many of CBRAIN's components makes the platform easily extensible. New data providers, execution servers, visualization tools and other components can be added to the platform with a minimal investment of time and effort. On a deeper level, a small investment in development time can extend the base data provider and SCIR APIs to allow compatibility with new types of storage and cluster management. As an example, our team has begun experimenting with the integration of Amazon's S3 cloud as a data provider. CBRAIN as a meta-scheduler does more than provide a uniform API to the heterogeneous scheduling of various sites; it handles maximum queue allocations, node vs. core scheduling, max load per node, specific environment variables, caches locations, and data transfer tools/protocols on a per site basis. The platform excels at bridging the gap in common standards between existing cyber-infrastructures, providing transparent access to grids, public HPC sites, and private infrastructure through a single common framework.

FUTURE WORK

We are prototyping methods to extend the job model to accommodate the provisioning of Virtual Machines (VMs) on HPC and Cloud infrastructure. Thanks to the flexible and integrated development of CBRAIN components, these extensions can reuse several of CBRAIN's core services. For instance, the meta-scheduler is used to launch VMs from disk images equipped with application tools which are simply stored on data providers and handled by the CBRAIN data management system. Executing tasks in VMs facilitates the deployment of tools on classic HPC clusters, enables the exploitation of clouds, and ensures a uniform computing environment across heterogeneous infrastructures. Deployed VMs are seen by the platform as computing sites, opening possibilities for finer cross-site load balancing. This increased mobility across traditional batch HPC sites and actual clouds will allow us to further leverage resources from these two types of services.

Moving forward, priorities for the platform include further development and refinement of the Web API to allow other systems to take advantage of the services offered by CBRAIN. There are plans to extend CBRAIN into fields other than neuroimaging, such as epigenomics and the humanities. The platform itself is generic, meaning that in principle it should be usable in any domain that requires computationally expensive processing of large data sets.

OBTAINING AND ACCESSING CBRAIN

The core CBRAIN codebase will be made available as an open source project in mid-2014. Please refer to the NITRC site for instructions (https://www.nitrc.org/projects/cbrain). Trial CBRAIN accounts can also be obtained upon registration (https://portal.cbrain.mcgill.ca). For any registration or source code access questions, our group can be contacted at cbrain-support.mni@mcgill.ca.

ACKNOWLEDGMENTS

This work has been funded by CANARIE, Canada's Advanced Research and Innovation Network (http://www.canarie.ca) and McGill University. We are grateful for the computing cycles, storage, and support obtained from Compute Canada (https://computecanada.ca) and our collaborators at KISTI (Korea) and at the Jülich Supercomputing Centre (Germany). In addition to the authors, we would like to thank Mathieu Desrosiers for contributing sustained development efforts and site project coordination.

REFERENCES

Amunts, K., Lepage, C., Borgeat, L., Mohlberg, H., Dickscheid, T., Rousseau, M. E., et al. (2013). BigBrain: an ultrahigh-resolution 3D human brain model. *Science* 340, 1472–1475. doi: 10.1126/science.1235381

Andreetto, P., Andreozzi, S., Avellino, G., Beco, S., Cavallini, A., Cecchi, M., et al. (2008). The gLite workload management system. *J. Phys. Conf. Ser.* 119, 062007. doi: 10.1088/1742-6596/119/6/062007

Andreetto, P., Andreozzi, S., Ghiselli, A., Marzolla, M., Venturi, V., and Zangrando, L. (2010). Standards-based job management in grid systems. *J. Grid Comput.* 8, 19–45. doi: 10.1007/S10723-010-9146-Z

Antoniu, G., Costan, A., Mota, B. D., Thirion, B., and Tudoran, R. (2012). A-brain: using the cloud to understand the impact of genetic variability on the brain. *ERCIM News* 89, 21–22. Available online at: http://ercim-news.ercim.eu/en89/

Bachle, M., and Kirchberg, P. (2007). Ruby on rails. *IEEE Softw.* 24, 105–108. doi: 10.1109/Ms.2007.176

Barbera, R., Donvito, G., Falzone, A., La Rocca, G., Milanesi, L., Maggi, G. P., et al. (2009). The GENIUS Grid Portal and robot certificates: a new tool for e-Science. *BMC Bioinformatics* 10(Suppl. 6):S21. doi: 10.1186/1471-2105-10-S6-S21

Baud, J.-P., Casey, J., Lemaitre, S., and Nicholson, C. (2005). "Performance analysis of a file catalog for the LHC computing grid," *IEEE International Symposium on High Performance Distributed Computing, 2005* (Research Triangle Park, NC), 91–99. doi: 10.1109/HPDC.2005.1520941

Bell, G., Hey, T., and Szalay, A. (2009). Computer science. Beyond the data deluge. *Science* 323, 1297–1298. doi: 10.1126/science.1170411

Brook, N., Bogdanchikov, A., Buckley, A., Closier, J., Egede, U., Frank, M., et al. (2003). "DIRAC - distributed infrastructure with remote agent control," in *Proceedings of the Computing in High Energy and Nuclear Physics* (La Jolla, CA), 1–8.

Buetow, K. H. (2005). Cyberinfrastructure: empowering a "third way" in biomedical research. *Science* 308, 821–824. doi: 10.1126/science.1112120

Catlett, C. (2002). "The philosophy of TeraGrid: building an open, extensible, distributed terascale facility," in *Proceeding of the 2nd IEEE International Symposium on Cluster Computing and the Grid* (Berlin), 8. doi: 10.1109/CCGRID.2002.1017101

Chervenak, A. L., Schuler, R., Ripeanu, M., Ali Amer, M., Bharathi, S., Foster, I., et al. (2009). The globus replica location service: design and experience. *IEEE Trans. Paral. Distrib. Syst.* 20, 1260–1272. doi: 10.1109/TPDS.2008.151

Costan, A., Tudoran, R., Antoniu, G., and Goetz, B. (2013). TomusBlobs: scalable data-intensive processing on Azure clouds. *Concur. Comput. Pract. Exp.* doi: 10.1002/cpe.3034. [Epub ahead of print].

Das, S., Zijdenbos, A. P., Harlap, J., Vins, D., and Evans, A. C. (2011). LORIS: a web-based data management system for multi-center studies. *Front. Neuroinform.* 5:37. doi: 10.3389/fninf.2011.00037

Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., et al. (2010). Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline. *PLoS ONE* 5:e13070. doi: 10.1371/journal.pone.0013070

Dinov, I. D., Van Horn, J. D., Lozev, K. M., Magsipoc, R., Petrosyan, P., Liu, Z., et al. (2009). Efficient, distributed and interactive neuroimaging data analysis using the LONI pipeline. *Front. Neuroinform.* 3:22. doi: 10.3389/neuro.11.022.2009

Ellert, M., Grønager, M., Konstantinov, A., Kónya, B., Lindemann, J., Livenson, I., et al. (2007). Advanced resource connector middleware for lightweight computational grids. *Future Gen. Comput. Syst.* 23, 219–240. doi: 10.1016/j.future.2006.05.008

Foster, I., and Kesselman, C. (2003). *The Grid 2: Blueprint for a New Computing Infrastructure*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Foster, I., Kesselman, C., Tsudik, G., and Tuecke, S. (1998). "A security architecture for computational grids," in *CCS '98 Proceedings of the 5th ACM Conference on Computer and Communications Security* (San Francisco, CA), 83–92. doi: 10.1145/288090.288111

Frey, J., Tannenbaum, T., Livny, M., Foster, I., and Tuecke, S. (2002). Condor-G: a computation management agent for multi-institutional grids. *Clust. Comput.* 5, 237–246. doi: 10.1023/A:1015617019423

Frisoni, G. B., Redolfi, A., Manset, D., Rousseau, M. E., Toga, A., and Evans, A. C. (2011). Virtual imaging laboratories for marker discovery in neurodegenerative diseases. *Nat. Rev. Neurol.* 7, 429–438. doi: 10.1038/nrneurol.2011.99

Glatard, T., Lartizien, C., Gibaud, B., da Silva, R., Forestier, G., Cervenansky, F., et al. (2013). A virtual imaging platform for multi-modality medical image simulation. *IEEE Trans. Med. Imaging* 32, 110–118. doi: 10.1109/TMI.2012.2220154

Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., et al. (2011). Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. *Front. Neuroinform.* 5:13. doi: 10.3389/fninf.2011.00013

Gronenschild, E. H. B. M., Habets, P., Jacobs, H. I. L., Mengelers, R., Rozendaal, N., van Os, J., et al. (2012). The effects of FreeSurfer version, workstation type, and Macintosh operating system version on anatomical volume and cortical thickness measurements. *PLoS ONE* 7:e38234. doi: 10.1371/journal.pone.0038234

Huedo, E., Montero, R. S., and Llorente, I. M. (2001). The GridWay framework for adaptive scheduling and execution on grids. *Scal. Comp. Pract. Exp.* 6, 1–8. doi: 10.12694/scpe.v6i3.332

Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M. (2012). Fsl. *Neuroimage* 62, 782–790. doi: 10.1016/j.neuroimage.2011.09.015

Jha, S., Kaiser, H., Merzky, A., and Weidner, O. (2007). "Grid interoperability at the application level using SAGA," in *IEEE International Conference on e-Science and Grid Computing* (Bangalore), 584–591. doi: 10.1109/E-SCIENCE.2007.39

Joshi, A., Scheinost, D., Okuda, H., Belhachemi, D., Murphy, I., Staib, L. H., et al. (2011). Unified framework for development, deployment and robust testing of neuroimaging algorithms. *Neuroinformatics* 9, 69–84. doi: 10.1007/s12021-010-9092-8

Kacsuk, P., Farkas, Z., Kozlovszky, M., Hermann, G., Balasko, A., Karoczkai, K., et al. (2012). WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities. *J. Grid Comput.* 10, 601–630. doi: 10.1007/s10723-012-9240-5

Karama, S., Bastin, M. E., Murray, C., Royle, N. A., Penke, L., Munoz Maniega, S., et al. (2013). Childhood cognitive ability accounts for associations between cognitive ability and brain cortical thickness in old age. *Mol. Psychiatry* 19, 555–559. doi: 10.1038/mp.2013.64

Kim, J. S., Singh, V., Lee, J. K., Lerch, J., Ad-Dab'bagh, Y., MacDonald, D., et al. (2005). Automated 3-D extraction and evaluation of the inner and outer cortical surfaces using a Laplacian map and partial volume effect classification. *Neuroimage* 27, 210–221. doi: 10.1016/j.neuroimage.2005.03.036

Kranzlmüller, D., Lucas, J. M., and Öster, P. (2010). "The european grid initiative (EGI)," in *Remote Instrumentation and Virtual Laboratories*, eds F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore (Springer US), 61–66. doi: 10.1007/978-1-4419-5597-5_6

Lavoie-Courchesne, S., Rioux, P., Chouinard-Decorte, F., Sherif, T., Rousseau, M. E., Das, S., et al. (2012). Integration of a neuroimaging processing pipeline into a pan-canadian computing grid. *J. Phys. Conf. Ser.* 341, 1–18. doi: 10.1088/1742-6596/341/1/012032

Lerch, J. P., Worsley, K., Shaw, W. P., Greenstein, D. K., Lenroot, R. K., Giedd, J., et al. (2006). Mapping anatomical correlations across cerebral cortex (MACACC) using cortical thickness from MRI. *Neuroimage* 31, 993–1003. doi: 10.1016/j.neuroimage.2006.01.042

Mackenzie-Graham, A. J., Van Horn, J. D., Woods, R. P., Crawford, K. L., and Toga, A. W. (2008). Provenance in neuroimaging. *Neuroimage* 42, 178–195. doi: 10.1016/j.neuroimage.2008.04.186

Markram, H. (2013). Seven challenges for neuroscience. *Funct. Neurol.* 28, 145–151. doi: 10.11138/FNeur/2013.28.3.144

Marru, S., Gardler, R., Slominski, A., Douma, A., Perera, S., Weerawarana, S., et al. (2011). Apache airavata. *Proc. ACM* 21, 21–28. doi: 10.1145/2110486.2110490

Olabarriaga, S. D., Glatard, T., and de Boer, P. T. (2010). A virtual laboratory for medical image analysis. *IEEE Trans. Inf. Technol. Biomed.* 14, 979–985. doi: 10.1109/TITB.2010.2046742

Park, S. M., and Humphrey, M. (2008). "Data throttling for data-intensive workflows," in *IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008* (Miami, FL), 1796–1806. doi: 10.1109/IPDPS.2008.4536306

Pordes, R., Petravick, D., Kramer, B., Olson, D., Livny, M., Roy, A., et al. (2007). The open science grid. *J. Phys. Conf. Ser.* 78, 012057. doi: 10.1088/1742-6596/78/1/012057

Redolfi, A., McClatchey, R., Anjum, A., Zijdenbos, A., Manset, D., Barkhof, F., et al. (2009). Grid infrastructures for computational neuroscience: the neuGRID example. *Future Neurol.* 4, 703–722. doi: 10.2217/fnl.09.53

Reuter, M., Schmansky, N. J., Rosas, H. D., and Fischl, B. (2012). Within-subject template estimation for unbiased longitudinal image analysis. *Neuroimage* 61, 1402–1418. doi: 10.1016/j.neuroimage.2012.02.084

Rex, D. E., Ma, J. Q., and Toga, A. W. (2003). The LONI pipeline processing environment. *Neuroimage* 19, 1033–1048. doi: 10.1016/S1053-8119(03)00185-X

Tröger, P., Rajic, H., Haas, A., and Domagalski, P. (2007). "Standardization of an API for distributed resource management systems," in *Seventh IEEE International Symposium on Cluster Computing and the Grid, 2007. CCGRID 2007* (Rio de Janeiro), 619–626. doi: 10.1109/CCGRID.2007.109

Van Horn, J. D., and Toga, A. W. (2013). Human neuroimaging as a "Big Data" science. *Brain Imaging Behav.* 8, 323–331. doi: 10.1007/s11682-013-9255-y

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 06 December 2013; accepted: 29 April 2014; published online: 21 May 2014.
Citation: Sherif T, Rioux P, Rousseau M-E, Kassis N, Beck N, Adalat R, Das S, Glatard T and Evans AC (2014) CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research. Front. Neuroinform. ***8****:54. doi: 10.3389/fninf.2014.00054*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 26 August 2014
doi: 10.3389/fninf.2014.00071

Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools

Dylan Wood[1]*, Margaret King[1], Drew Landis[1], William Courtney[1], Runtang Wang[1], Ross Kelly[1], Jessica A. Turner[1,2] and Vince D. Calhoun[1,3]

[1] The Mind Research Network and LBERI, Albuquerque, NM, USA
[2] Department of Psychology, Georgia State University, Atlanta, GA, USA
[3] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Fred Prior, Washington University School of Medicine, USA
Benjamin Thyreau, Tohoku University, Japan

***Correspondence:**
Dylan Wood, The Mind Research Network, 1101 Yale Blvd NE, Albuquerque, NM 87131, USA
e-mail: dwood@mrn.org

Neuroscientists increasingly need to work with *big data* in order to derive meaningful results in their field. Collecting, organizing and analyzing this data can be a major hurdle on the road to scientific discovery. This hurdle can be lowered using the same technologies that are currently revolutionizing the way that cultural and social media sites represent and share information with their users. Web application technologies and standards such as RESTful webservices, HTML5 and high-performance in-browser JavaScript engines are being utilized to vastly improve the way that the world accesses and shares information. The neuroscience community can also benefit tremendously from these technologies. We present here a web application that allows users to explore and request the complex datasets that need to be shared among the neuroimaging community. The COINS (Collaborative Informatics and Neuroimaging Suite) Data Exchange uses web application technologies to facilitate data sharing in three phases: Exploration, Request/Communication, and Download. This paper will focus on the first phase, and how intuitive exploration of large and complex datasets is achieved using a framework that centers around asynchronous client-server communication (AJAX) and also exposes a powerful API that can be utilized by other applications to explore available data. First opened to the neuroscience community in August 2012, the Data Exchange has already provided researchers with over 2500 GB of data.

Keywords: open neuroscience, big data, neuroinformatics, data sharing, query builder, javascript

INTRODUCTION

Many of the questions faced by the human neuroimaging community can no longer be answered through studying small data sets due to the wide structural and functional variance between individual subjects. Instead, neuroimaging researchers need to look at large populations in order to accurately distinguish between overarching trends and individual outliers. Accumulating such large data sets can be time consuming and expensive—often prohibitively so. In response to this challenge, some members of the neuroimaging community are molding a new approach to data collection. This new approach has been dubbed Open Neuroscience, and it necessitates that individual researchers will openly share phenotypic, genotypic and neuroimaging data and collection methodologies (Milham, 2012).

Thus far, several large datasets and sharing platforms have been released in the spirit of the Open Neuroscience initiative with great support and success. One of the earliest examples was the fMRI Data Center (fMRIDC), which consolidated and shared thousands of datasets from 2000 to 2007 (Van Horn et al., 2001; Van Horn and Gazzaniga, 2013). Later came the 1000 Functional Connectomes Project (FCP), which released a curated dataset of 1300 subjects in December 2009[1]. Other recent examples of curating and centralizing multi-site data for open distribution include the Biomedical Informatics Research Network (BIRN), the Functional Biomedical Informatics Research Network (F-BIRN) and The International Neuroimaging Data-sharing Initiative (INDI). INDI hopes to expand on the success of the FCP project by focusing on establishing strong phenotypic datasets to accompany the imaging data[2].

All of the approaches mentioned thus far have utilized a curation process in which data is manually checked for quality and adherence to project-specific data collection and processing standards. An alternative approach is seen in XNAT Central and the National Database for Autism Research (NDAR), which are centralized repositories for researchers to deposit data[3] (Hall et al., 2012). Other researchers may then analyze and download the posted neuroimaging datasets. Data that is deposited in these databases is openly available to the community, and therefore must be fully anonymized before upload. XNAT Central does not rely on manual curation to ensure quality and standards, and places the burden of data-verification on the downloader.

[1] https://fcon_1000.projects.nitrc.org/ accessed 1/28/2014

[2] http://fcon_1000.projects.nitrc.org/indi/docs/INDI_MISSION_STATEMENT.pdf

[3] XNAT Central https://central.xnat.org/

Here we propose another approach to providing an Open Neuroscience sharing infrastructure. The proposed approach does not require manual curation by a centralized organization, yet promises stricter adherence to standards than a completely open approach. The key to this approach is a neuroinformatics data management platform called the Collaborative Informatics and Neuroimaging Suite (COINS) (Scott et al., 2011). Researchers have noted the importance of managing data within an informatics platform from the time of collection onward (Mennes et al., 2013). By doing so, data is stored according to generalizable ontologies that can be mapped across studies, sites and even between data management platforms. In addition to offering a framework to organize data for universal mapping, COINS allows researchers to store all of their research data in one place, and then selectively (or globally) share that data in a manner that satisfies the Health Information Privacy and Accountability Act (HIPAA) and secures Protected Health Information (PHI) against accidental exposure. The COINS Data Exchange (DX) is a vehicle for the greater research community to explore, request and download this shared data. The following text outlines the technology used by DX as well as an analysis of the success of the system.

MATERIALS AND METHODS

The COINS Data Exchange (DX; http://coins.mrn.org/dx) was designed to be a repository where researchers from all over the world can intuitively share data. DX performs two main processes: The first provides an intuitive interface through which researchers can explore, request and download data stored within the COINS database. The second allows researchers who are not already storing their data in COINS to upload that data for sharing. This paper will focus on the first step of the first process: data exploration.

DX uses a unique exploratory interface to visually construct ad-hoc queries. The interface consists conceptually of a single workspace that represents a *request*. The *request* can have one or more logical *groups*, each of which can have zero or more child *groups* and zero or more *filters*. The *groups* define the logical relationship (and vs. or) between individual children of that group. The workspace is populated with groups and filters by clicking and dragging elements on the screen into the workspace. In addition, filters may be converted to templates by super users, and those templates can be used as a starting point for other users looking for similar data. An example of the interface, which is called the Data Catalog, is shown in **Figure 1**.

The Data Catalog is made possible by modern web application technologies such as JavaScript and JQuery, HTML5 APIs and CSS3. The development of the Data Catalog was expedited by utilizing Node.js, which facilitated the reuse of libraries in the client and the server (Tilkov and Vinoski, 2010). Historically, web-based applications have been created using one server-side programming language (e.g., java, ruby, php), and a completely different client-side language (JavaScript). Node.js is a paradigm shift from this methodology in that it allows developers to use JavaScript on both the server- and client-side. This code reuse allows developers to program much more efficiently. Other notable companies also using node.js are PayPal, Groupon, eBay, and LinkedIn. PayPal estimates that they were able to create new features twice as fast with fewer people, use 33% fewer lines of code, and generate 40% fewer files when they used node.js as compared to their previous methodology utilizing disparate client and server languages[4].

ARCHITECTURE

The interface is delivered to the browser in the form of a HTML web page dynamically generated by PHP scripts and several javascript libraries. All files are served from the same Linux-Apache-PostgreSQL-PHP (LAPP) servers that host other COINS web applications. Since the Data Catalog is accessed within the COINS web application, security is managed by the COINS Central Authentication System. After a user logs in, their PHP session information is stored on a centralized memcached server, which is accessible to the COINS web application servers as well as the Node.js servers that host the Data Catalog web services (Brad, 2004; Olson et al., 2014). Once the Data Catalog interface has been loaded into the browser, all data queries will be sent to a separate webservice running on a Node.js server. This is illustrated in **Figure 2**.

The components of the Data Catalog workspace mentioned above, and shown in **Figure 1** are represented as JavaScript objects that are defined in the libraries used by both the browser and Node.js server. This allows for objects that represent the user's query (discussed below) to be easily passed from the server to the client and back. This communication is handled via standard asynchronous HTTP(S) requests, and can also be leveraged by other automated services (also shown in **Figure 2**). For instance, the NIH-Funded SchizConnect Data Federation is working with COINS and XNAT Central and the Human Imaging Database (HID) to create a tool that will automatically compile a comprehensive catalog of data available on both sharing resources. This tool retrieves information about data available in DX via a RESTful API[5] (SchizConnect Data Federation).

USER INTERFACE INITIALIZATION

When the user interface of the Data Catalog is first initialized, a new Request object is constructed. As part of the construction, the top-group Group object is also constructed, and the filterable modalities are asynchronously retrieved from the server and loaded into the *modalities* property of the new Request object from the server. Each modality represents a mode of data for which there is at least one filterable attribute, and for which statistics should be calculated and displayed.

When the request is modified (either by assigning it a label, or adding a new Filter or Group object, it will be persisted to the server. This is done by calling the Request object's write() method. The write() method utilizes the Request object's toJSON() method, which in-turn calls the toJSON() methods of all child objects (Groups and Filters) in order to properly serialize them. The JSON representation of the object is then sent to the server via a POST HTTP request. On the server, a new Request is once again constructed using the same library that was used on the client. Next, the new Request object's fromJSON() method is

[4] https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/ Dec 23, 2013

[5] http://niacal.northwestern.edu/projects/18

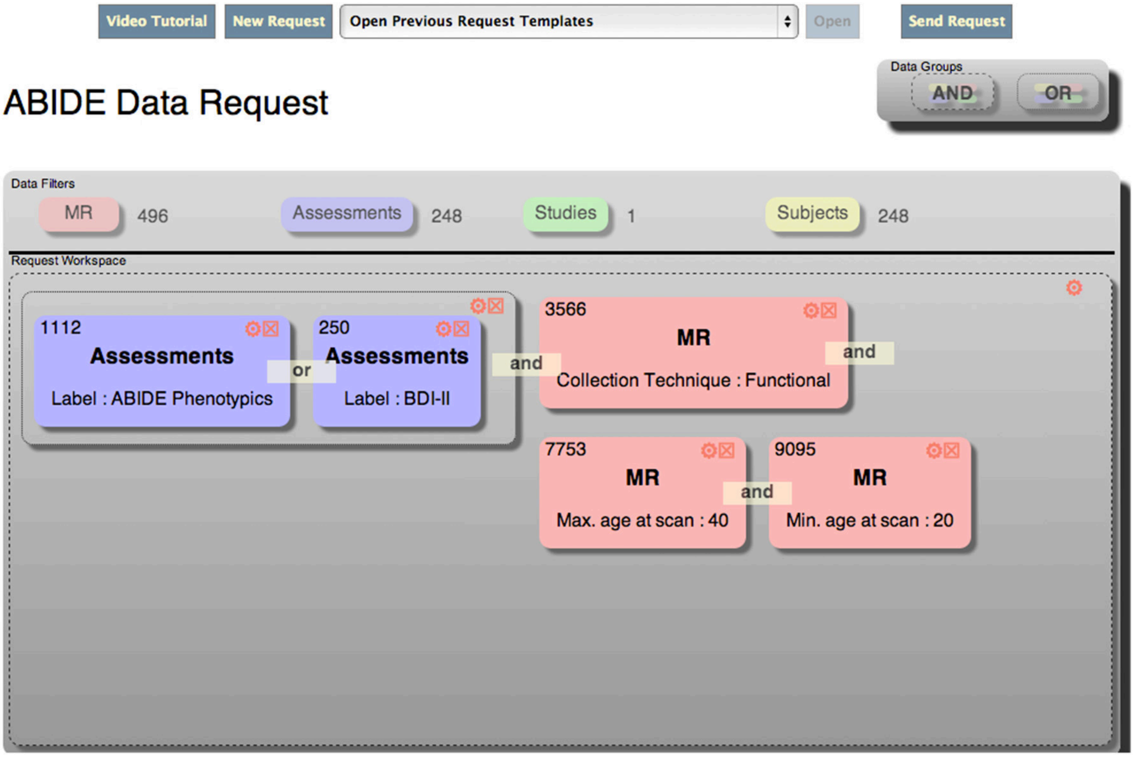


FIGURE 1 | DX data catalog exploratory filtering tool.

invoked, which in-turn recursively calls the fromJSON() methods of each child object to properly unserialize all objects. Following unserialization, the write() method of the server-side Request object is invoked, which overloads the write() method defined in the shared definition of Request. This method writes the relevant properties of the Request to the database, setting the id property to the value assigned by the database. Write() also recursively calls the write() methods of all child objects, so that they are also persisted to the database and assigned identifiers accordingly. Finally, the Request is once-again serialized to JSON, and sent back to the client, where it is unserialized and replaces the current Request object before being rendered.

As filters and groups are added or modified, objects representing those entities are created or updated on the client. Those objects are then encoded into JavaScript Object Notation (JSON) strings and sent to the server for processing via asynchronous HTTP(S) requests (Bray, 2014). The server then parses the JSON strings into proper objects, and forms SQL queries to retrieve statistics about the objects from the COINS database. The resulting statistics are then appended as properties of the objects before JSON encoding them and sending them back to the client where they are used to update the interface with statistics about the current request.

JAVASCRIPT DATA MODEL

A simplified model of the Javascript objects that comprise the Data Catalog is shown in **Figure 3**. Some properties were excluded from the model for clarity. Each object's prototype encapsulates a method to render itself in HTML: a functionality only used on the client. The prototype for each object also contains methods to deconstruct itself into a JSON string that can be sent across a wire. Similarly, each prototype has a method to reconstruct itself from a JSON string or standard object. These methods are employed on both the client and the server to facilitate passing the objects back and forth. Each of the objects illustrated in **Figure 3** are explained in more detail below.

The Request object is the top-level object for the Data Catalog UI, and as such, it contains pointers to all other objects relevant to the UI. When a new, blank request is first started by a user, it is assigned a unique identifying integer, which is recorded in the server-side database, and assigned as the id property for the request. Another property of the Request object is populated upon initialization: modalities. Each modality is an object which specifies the type of data for which metrics are to be displayed, and for which filters should be available. A user-specified label may also be associated with the request, and will be persisted to the server-side database as well.

The topGroup property of the Request is populated upon request initialization, and points to a *group* object. Each group object also has a server-defined identifier (id), and properties to list child other groups and filters that reside within the current group. Additionally, groups have a *type* property which can be either "and" or "or."

Groups may contain zero or more *filter* objects. Filter objects contain a list of attributes, which correspond to rows of the *dx_source_attributes* table mentioned elsewhere in this paper. At

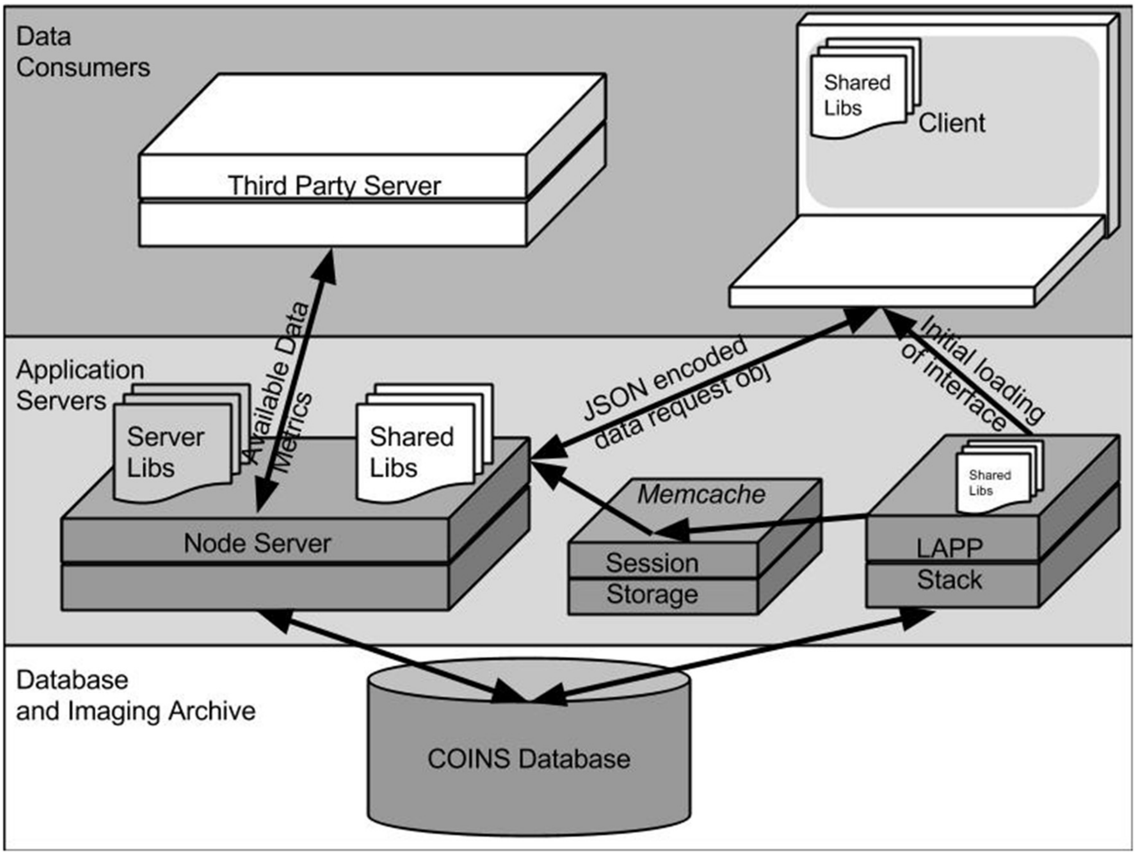


FIGURE 2 | COINS DX infrastructure.

present, the user interface only supports one attribute per filter, however, the attributes property is an array in anticipation of future changes. Other properties of each filter object includes the type and statistics associated with the modality of the filter, and a server-issued identifier, which corresponds to a persistent representation of the filter in the COINS database.

Filter *attributes* have properties that define the *source_attribute* to which the property corresponds, as well as the value and its description, as selected by the user for that attribute (*optionId* and *optionDesc*, respectively).

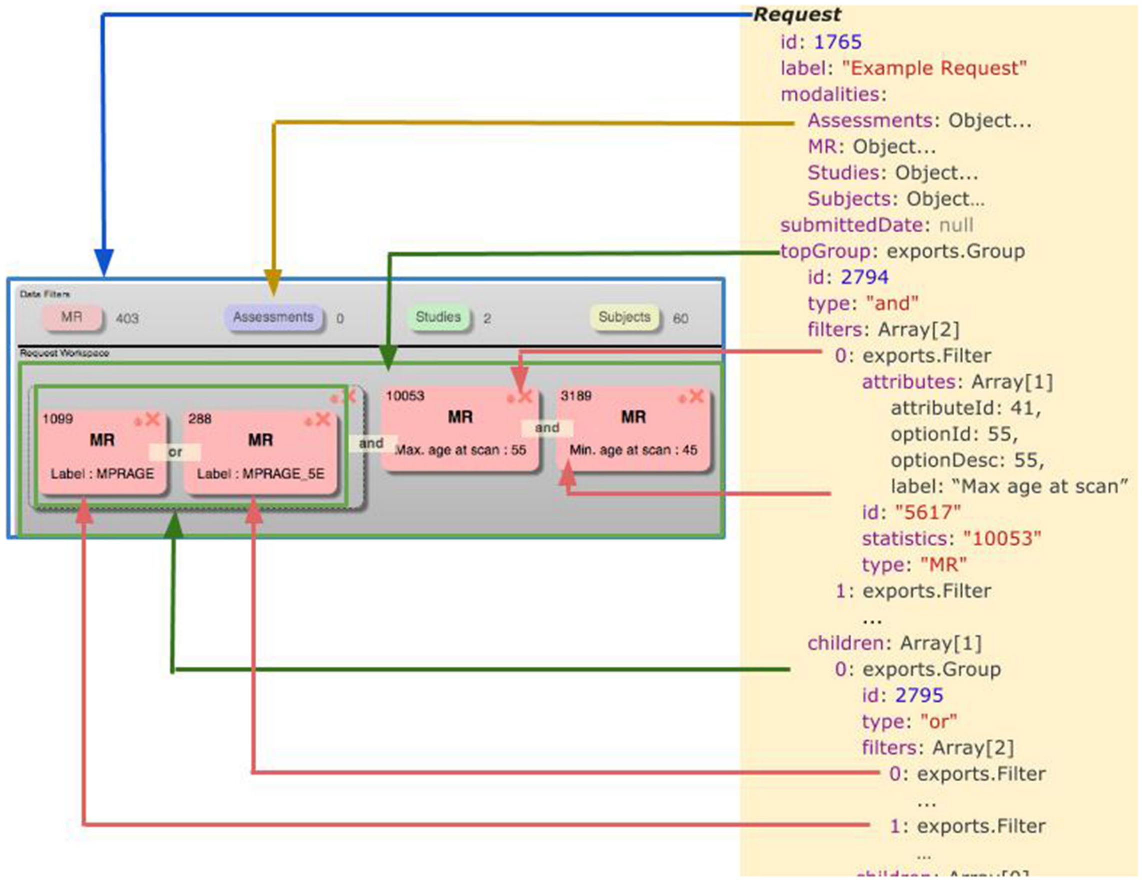
SERVER-SIDE PROCESSING

Server-side-only libraries extend the object prototypes in order to add database-related functionality. For example, each server-side object prototype (e.g., ServerRequest, ServerGroup, ServerFilter) exposes a method to persist a representation of each object to the database for persistence. Other server-side-only methods generate PL/SQL code to process statistics or metadata about the object in the database. In the case of a filter object, the PL/SQL code inserts the primary keys of all data that matches the filter's *filterAttributes* into a temporary table where it can be intersected or unioned with other filters' data (depending on what type of group the filter is in). As with the JSON (un)serialization methods, all aforementioned methods call their correlate-methods of all child objects (Request.render() will call Group.render() for all Groups in the request, and so on).

MODALITIES AND FILTERS

When a request is first initiated in the client, a list of available modalities and filters is retrieved from the database. These data are manually curated by modifying data stored in the COINS database. **Figure 4** depicts the tables discussed in this paper, and a more general understanding of the COINS database was published in 2010 by Scott et al. (2011). The modalities are populated from a table in the database, which consists of modality labels and pointers to the tables and primary keys that they correspond to. Statistics displayed for each modality are calculated by tallying the number of unique primary key values are matched by the user's query. For example, the Study modality corresponds to the a materialized view of available studies and the *anonymization_ids* of subjects that are enrolled in them (*dx_studies_mv*), and the *study_id* primary key.

Filters for each modality are also configured and stored in the COINS database. The table *mrs_source_attributes* stores available attributes which can be filtered upon. Each attribute is linked to a modality via a foreign key constraint. Other columns of mrs_source_attributes specify which columns of the *modality's table* should be used for the available values and value-descriptions available for each filterable attribute. Continuing



```
Request
  id: 1765
  label: "Example Request"
  modalities:
    Assessments: Object...
    MR: Object...
    Studies: Object...
    Subjects: Object...
  submittedDate: null
  topGroup: exports.Group
    id: 2794
    type: "and"
    filters: Array[2]
      0: exports.Filter
        attributes: Array[1]
          attributeId: 41,
          optionId: 55,
          optionDesc: 55,
          label: "Max age at scan"
        id: "5617"
        statistics: "10053"
        type: "MR"
      1: exports.Filter
        ...
    children: Array[1]
      0: exports.Group
        id: 2795
        type: "or"
        filters: Array[2]
          0: exports.Filter
            ...
          1: exports.Filter
            ...
        children: Array[0]
```

FIGURE 3 | Simplified model of a data catalog request.

with the previous example of the Study modality, the value and value-description properties of the *Study Label* filter-attribute correspond to the *label* and *description* columns of *mrs_studies*.

QUERY GENERATION

Among the methods exposed by the server-side-only libraries are methods to convert each filter's attributes into SQL queries. Server-side *filter* objects expose a method *generateSQL(), which* generates SQL to select the identifiers of data that is matched by each filter-attribute's "optionId" and "operator." Similarly, the *group* object exposes a *generateSQL()* method that will *union* (type = *or*) or *intersect* (type = *and*) the modular queries created by the group's filters and child-groups.

In order to allow groups to contain filters of disparate modalities, some additional logic is necessary. SQL modules generated by child-groups and filters of the same modality should be combined using the modality's primary key (e.g., *subjects with age* ≥ *25* AND *subjects with age* ≤ *55*). SQL modules generated by child-groups and filters of varying modalities must be combined using the subject-anonymization-identifier (e.g., *subjects with age* ≥ *25* AND *MR with series label* = "*MPRAGE*").

This additional logic is assisted by automatically redrawing user-defined groupings every time the user modifies the request object. The re-drawing looks for groups that contain three or more filters, where at least two of which are of the same modality and at least one of which is of a different modality than the others. These filters are then split up into sub-groups according to their modalities: for instance, an "and" group containing two *subject* filters and two *MR* filters will be redrawn to contain two child groups: one for *subject* filters and one for *MR* filters.

The *generateSQL()* methods are called for each object by the object's parent (i.e., The *request* object calls *topGroup.generateSQL()*, which in turn calls the *generateSQL()* method of each of its child groups and filters, and so on). When all method calls have returned, the request receives a single SQL statement that will yield the subject-anonymization-identifiers and modality-specific-primary-keys of all data that is matched by the request. Additionally, the SQL generated by each object can be run independently to retrieve statistics about the amount of data matched by that object.

A note about security

Whenever utilizing client-generated values to generate SQL, it is important to screen for SQL injection attacks. The data catalog implements the same security measures practices elsewhere in the COINS application. First, the login-role used by the application does not have read or write access to underlying tables that contain data. Instead, all database reads are performed by selecting

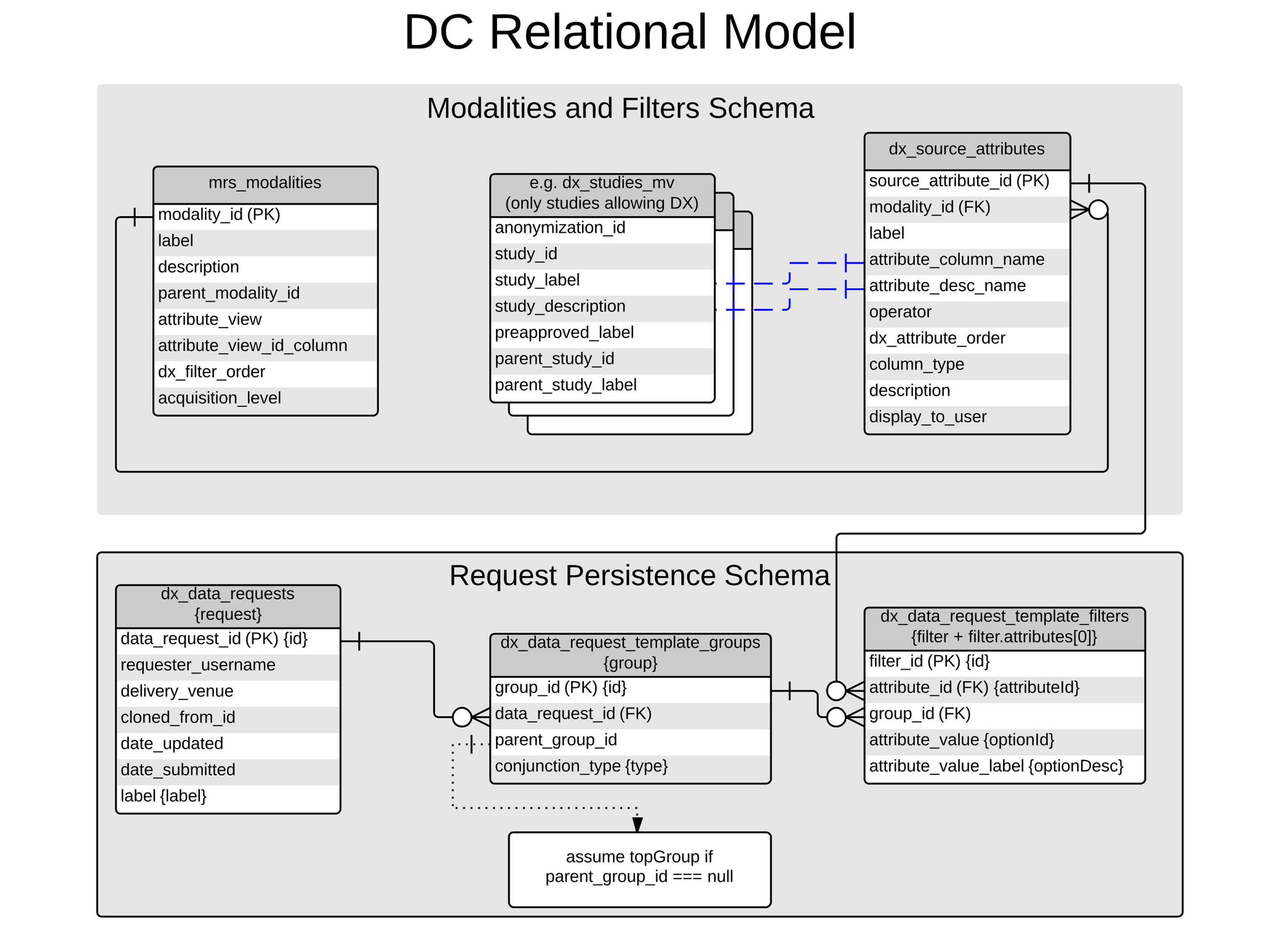


FIGURE 4 | Database schema for data catalog.

data from views, rather than directly from tables. Similarly, all write operations are performed by calls to stored functions. In the case of the data catalog application, the login-role's read access is restricted to data-catalog related views and functions that persist user's requests to the database. Thus, any SQL injection attacks would not reveal any more information than is readily available through the user interface. For the sake of added caution, other standard protections are also implemented, such as type-checking, query parameter binding, user authentication, a 30-min logout window, and record-modification history logging.

RESULTS

APPLICATION AND FEATURES

The Data Catalog is a critical component of the COINS Data Exchange. It allows users to construct complex ad-hoc queries against sharable data in the COINS database in an exploratory way to form a request for data. After constructing a request, the request can be submitted, which will notify all COINS users that own the data being requested that some of their data is being requested. The submitted request can be accepted or denied by the data owners after the requester and owner have exchanged messages through the integrated messaging system. All messages are stored indefinitely, and can be used as official documents or an audit trail if needed. If one data-owner approves the request, and another denies it, only the approved subset of data will be made available to the requester. Data associated with accepted requests is packaged and zipped on the COINS servers, and the requester is notified when the package(s) are ready for download. The packaging, zipping, and download process is also quite interesting, but will not be explained in detail here.

INTEGRATION WITH COINS

Data collected via COINS is easily shareable in DX. Study administrators are provided very fine-grained control over which data is shared: individual subject types, subjects, scans, instruments or assessments may be excluded or included. Additionally, sharing benefits from the centralized approach of COINS. Studies that have collected data using shared instruments can now expose their data to sharing more easily. This allows a Data Catalog user to request data from two studies that have collected data using the Balanced Depression Inventory II (BDI-II) with a single filter (*Instrument label* = "BDI-II").

DATA SHARED

The first publicly-available dataset to be shared on COINS DX was the[6] ABIDE dataset. Released in the COINS DX on August 30th 2012, the ABIDE dataset consists of functional and structural imaging and phenotypic data from more than 1000 participants gathered from 15 different sites around the world (Nooner et al., 2012). The ABIDE dataset was imported into the COINS data management system and made available for sharing. To date, 166 individuals have downloaded over 2450 GB of ABIDE data through COINS DX.

Next, the NKI Rockland Sample made their first data release in March 2013[7]. Unlike the ABIDE dataset, the NKI Rockland Sample dataset was collected directly using COINS. This allows the NKI research team to make periodic releases by simply selecting which subjects and data is ready to be shared in DX. Their changes are reflected instantly in the Data Catalog. Researchers requesting access to the NKI Rockland Sample dataset require individual approval after agreeing to a DUA. Despite the more rigorous approval process, over 1200 GB of data have been approved for sharing and downloaded by 15 researchers from around the world.

DISCUSSION

The COINS Data Catalog harnesses modern web technologies to extend a popular neuroinformatics platform for use in the context of Open Neuroscience. The architecture of the application has proven flexible, maintainable, and secure. Moreover, two large datasets have been successfully shared on an international scale. One of those datasets was collected and compiled outside of COINS, then successfully imported. The other dataset is part of an ongoing collection effort using COINS tools, and can be easily curated by the data owners. Over all, over 3500 GB of data have been shared through the COINS Data Exchange since September 2012.

There remains a huge potential to share an increasing amount of data using DX: There are currently over 500 studies being managed with COINS. These studies have collected 342,000 clinical assessments and 31,400 MRI and MEG scan sessions from 22,100 participants at many sites across the United States including The Mind Research Network, Nathan Kline Institute, University of Colorado—Boulder, Olin Neuropsychiatry Research Center (King et al., 2014). Each of these studies can easily elect to allow some or all of their data to be explored and requested through DX.

As more studies elect to share their data through DX, the greater the number of filtering options will become during data exploration. If the number of filtering options grows too large, it may become difficult for a researcher to locate the options that apply to their own interests. It is important therefore to create data dictionaries and ontological mappings for the large amount of data currently stored within COINS. Such mappings will allow for multi-level filtering options that correspond to other popular common data elements.

Looking ahead, the developers of the COINS DX are excited to implement more features to aid sharing within the Open Neuroscience community. Dynamic requests are being developed, which will periodically alert researchers if new data is made available which matches one of their existing filters. Further improved API performance and documentation is on the way, and will aid in integration with projects such as SchizConnect (SchizConnect Data Federation) and Neurodebian[8].

ACKNOWLEDGMENT

This work was supported in part by NIH grants 2R01EB005846, P20GM103472 and 1U01MH097435.

REFERENCES

Brad, F. (2004). Distributed caching with memcached. *Linux J.* 124, 5–10.

Bray, T. (2014). *The JavaScript Object Notation (JSON) Data Interchange Format Internet Engineering Task Force (IETF), 7159*, ISSN: 2070–1721.

Hall, D., Huerta, M. F., McAuliffe, M. J., and Farber, G. K. (2012). Sharing heterogeneous data: the national database for autism research. *Neuroinformatics* 10, 331–339. doi: 10.1007/s12021-012-9151-4

King, M. D., Wood, D., Miller, B., Kelly, R., Landis, D., Courtney, W., et al. (2014). Automated collection of imaging and phenotypic data to centralized and distributed data repositories. *Front. Neuroinform.* 8:60. doi: 10.3389/fninf.2014.00060

Mennes, M., Biswal, B. B., Castellanos, F. X., and Milham, M. P. (2013). Making data sharing work: the FCP/INDI experience. *Neuroimage* 82, 683–691. doi: 10.1016/j.neuroimage.2012.10.064

Milham, M. P. (2012). Open neuroscience solutions for the connectome-wide association era. *Neuron* 73, 214–218. doi: 10.1016/j.neuron.2011.11.004

Nooner, K. B., Colcombe, S. J., Tobe, R. H., Mennes, M., Benedict, M. M., Moreno, A. L., et al. (2012). The NKI-Rockland sample: a model for accelerating the pace of discovery science in psychiatry. *Front. Neurosci.* 6:152. doi: 10.3389/fnins.2012.00152

Olson, P., Achour, M., Betz, F., Dovgal, A., Lopes, N., and Magnusson, H. (2014). *PHP Manual. PHP Documentation Group*. Available Online at: http://www.php.net/docs.php, accessed May 20145

Scott, A., Courtney, W., Wood, D., de la Garza, R., Lane, S., King, M., et al. (2011). COINS: an innovative informatics and neuroimaging tool suite built for large heterogeneous datasets. *Front Neuroinform* 5:33. doi: 10.3389/fninf.2011.00033

Tilkov, S., and Vinoski, S. (2010). Node.js: using javascript to build high-performance network programs. *IEEE Internet Comput.* 14, 80–83. doi: 10.1109/MIC.2010.145

Van Horn, J. D., Grethe, J. S., Kostelec, P., Woodward, J. B., Aslam, J. A., Rus, D., et al. (2001). The Functional Magnetic Resonance Imaging Data Center (fMRIDC): the challenges and rewards of large-scale databasing of neuroimaging studies. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 356, 1323–1339. doi: 10.1098/rstb.2001.0916

Van Horn, J. D., and Gazzaniga, M. S. (2013). Why share data? Lessons learned from the fMRIDC. *Neuroimage* 82, 677–682. doi: 10.1016/j.neuroimage.2012.11.010

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 03 February 2014; accepted: 23 July 2014; published online: 26 August 2014.
Citation: Wood D, King M, Landis D, Courtney W, Wang R, Kelly R, Turner JA and Calhoun VD (2014) Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools. Front. Neuroinform. ***8****:71. doi: 10.3389/fninf.2014.00071*
This article was submitted to the journal Frontiers in Neuroinformatics.

[6] http://fcon_1000.projects.nitrc.org/indi/abide/

[7] http://neuro.debian.net/

[8] http://neuro.debian.net/

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 23 April 2014
doi: 10.3389/fninf.2014.00032

Integrated platform and API for electrophysiological data

Andrey Sobolev, Adrian Stoewer, Aljoscha Leonhardt, Philipp L. Rautenberg, Christian J. Kellner, Christian Garbers and Thomas Wachtler*

Department Biology II, German Neuroinformatics Node, Ludwig-Maximilians-Universität München, Planegg, Germany

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Sam Neymotin, State University of New York, USA
Daniel K. Wójcik, Polish Academy of Sciences, Poland

****Correspondence:***
Andrey Sobolev, Department Biology II, German Neuroinformatics Node, Ludwig-Maximilians-Universität München,
Grosshaderner Str. 2,
82152 Martinsried, Planegg,
Germany
e-mail: sobolev@bio.lmu.de

Recent advancements in technology and methodology have led to growing amounts of increasingly complex neuroscience data recorded from various species, modalities, and levels of study. The rapid data growth has made efficient data access and flexible, machine-readable data annotation a crucial requisite for neuroscientists. Clear and consistent annotation and organization of data is not only an important ingredient for reproducibility of results and re-use of data, but also essential for collaborative research and data sharing. In particular, efficient data management and interoperability requires a unified approach that integrates data and metadata and provides a common way of accessing this information. In this paper we describe GNData, a data management platform for neurophysiological data. GNData provides a storage system based on a data representation that is suitable to organize data and metadata from any electrophysiological experiment, with a functionality exposed via a common application programming interface (API). Data representation and API structure are compatible with existing approaches for data and metadata representation in neurophysiology. The API implementation is based on the Representational State Transfer (REST) pattern, which enables data access integration in software applications and facilitates the development of tools that communicate with the service. Client libraries that interact with the API provide direct data access from computing environments like Matlab or Python, enabling integration of data management into the scientist's experimental or analysis routines.

Keywords: electrophysiology, data management, neuroinformatics, web service, collaboration, neo, odml

1. INTRODUCTION

1.1. DATA MANAGEMENT IN ELECTROPHYSIOLOGY—COSTS, BENEFITS, AND NEEDS

Advances in technology and methodology during the past years have dramatically increased the volume and complexity of data recorded in electrophysiological experiments. At the same time, progress in neuroscience increasingly depends on collaborative efforts, exchange of data, and re-analysis of previously recorded data. Thus, ensuring that data stays accessible, that data processing is reproducible, and that data can be shared and re-used has become a challenge for many laboratories (Herz et al., 2008).

Obstacles to efficient data management arise not only from the variety of data formats and constraints of accessing data in proprietary formats, but also from the amount and complexity of additional information about the experiment that needs to be collected and stored. This additional information, which is commonly called "metadata" despite the fact that it is to large part data supplementing the recorded data (**Figure 1**), is not only necessary to reproduce the study but also essential for searching, selecting, and analyzing the data.

Collecting and storing metadata comprehensibly together with the recorded data is also a facilitating requisite for sharing the data. Data sharing starts in the lab, where data needs to stay accessible and understandable for the experimenter even years after the study, and lab members need to be able to find and access data even after the person that performed the experiment has left the lab. In collaborations with scientists outside the laboratory, data need to be selected and the collaborators need to be able to understand the data. Having a data organization in the lab where all data and metadata is kept together in defined formats and organized structure can reduce both the experimenter's work for data preparation and the collaborator's efforts to read and understand the data. In the same way, efficient data organization minimizes the time and work necessary for preparing data to make it generally accessible, thus reducing the barriers to public sharing and data publication.

Experimental metadata typically have to be collected from various sources and in different formats—different measurement devices, software code, notes entered during the experiment, etc.—and have to be brought into compatible formats, which can require considerable effort. Typically, each lab defines its own methods, procedures, and format conventions for organizing and managing the data. If common tools and formats were available, workload and time demand in the labs would be reduced and data exchange would require less effort and time.

Developing common tools and standardized formats has turned out to be particularly challenging for the area of electrophysiology (Teeters et al., 2013). This field faces an enormous variety in experimental methodology, with a large number of data acquisition systems, file formats that are often vendor-specific and undocumented (Garcia et al., 2014), a variety of electrode configurations, species, preparations, stimuli, and overall experimental paradigms. Currently, common organization schemes or standards for accessing data do not exist. Thus, for data exchange, often substantial work is necessary to make the data accessible in one form or another. Moreover, in electrophysiology the

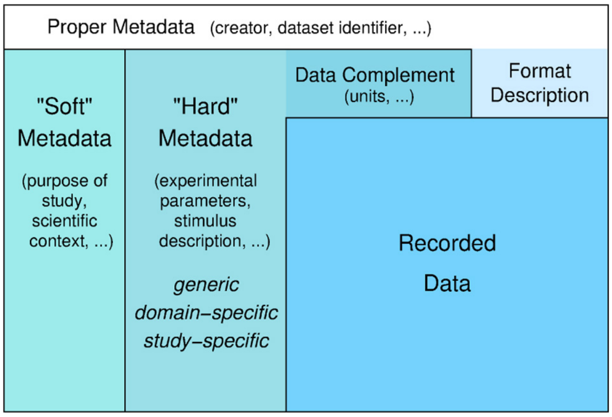

Proper Metadata (creator, dataset identifier, ...)
"Soft" Metadata
(purpose of study, scientific context, ...)
"Hard" Metadata
(experimental parameters, stimulus description, ...)
generic
domain–specific
study–specific
Data Complement (units, ...)
Format Description
Recorded Data

FIGURE 1 | Levels of (meta)data. Recorded data and additional information that is necessary for understanding and appropriate analysis of the data. Information about the format in which the data are stored is required to read the data. Information that complements the raw stored numbers, such as sampling rate, scaling factors, units, is required to understand the data as measured signals. To meaningfully analyze the data, information about the experimental context is necessary, like conditions of preparation, stimulation, etc. This information in principle can be formalized and stored in machine-readable form ("hard metadata") so that it can be used for data selection and analysis. This metadata can be further categorized into generic, domain-specific, and study-specific information. "Soft metadata" is the information about the overall scientific context and aim of the study, reasons for choosing certain parameters, etc., for which currently we have no way of formalizing or machine-processing. The distinction between supplementing data and proper metadata is to some degree arbitrary. For example, the date when an experiment was performed might usually be considered as proper metadata. However, in some analysis the time between experiments might be an important parameter to be taken into account. In this case, the date of the experiments can be used as data in the analysis to determine this information.

experimental variety and complexity results in corresponding variety and complexity of the metadata. While a set of minimal common metadata for a neuroscience experiment has been proposed (Gibson et al., 2008), for each dataset further specific information needs to be provided. As long as a comprehensive ontology for this field is missing (Bandrowski et al., 2013), approaches to achieve a common scheme for metadata description must leave sufficient flexibility to account for the variety and heterogeneity of experiments (Grewe et al., 2011).

1.2. DATABASES AND SHARING PLATFORMS FOR ELECTROPHYSIOLOGICAL DATA

In the past years, several initiatives to support data sharing in neurophysiology have emerged. One of the first public databases for electrophysiological data was the neurodatabase.org[1] project (Gardner, 2004). In this project, an elaborate data model and format along with a query protocol for the exchange of neurophysiological data were developed. The data, typically obtained from publications, is made available with extensive metadata and provided in a format specifically developed for this project.

The SenseLab Project[2] is a long-term initiative to build a repository for multidisciplinary models of neurons and neural systems (Crasto et al., 2007). It is a part of the Neuroscience Information Framework[3] (NIF, Gardner et al., 2008) and the International Neuroinformatics Coordinating Facility (INCF) [4]. The project provides open databases (ModelDB, NeuronDB, etc.) designed for certain aspects like neural modeling, neural cell properties, modeling of neurocircuits and several others.

The CRCNS.org site[5] hosts electrophysiological data that have been specifically selected by contributing labs for the purpose of making the data available to the public (Teeters et al., 2008). Typically, these data are from published studies and have been made available for re-use. Data format, annotation and documentation are different for each dataset.

The CARMEN project[6] provides a platform for data analysis and data exchange where the owner of the data can keep the data private, or can make the data available to selected users or the public. The platform also provides services for data analysis (Austin et al., 2011). For this purpose Carmen has introduced an internal file format, Carmen NDF[7], that is suitable for storing electrophysiological and other types of neuroscientific data. The user has the option to enter metadata describing the experiment in which the data were recorded. This is done via web forms that provide fields corresponding to the minimal metadata that were proposed by the Carmen consortium (Gibson et al., 2008).

The German Neuroinformatics Node (G-Node) provides a platform for data organization and data sharing of neurophysiological data[8]. Users can upload, organize, and annotate their data, and make them accessible to selected users or the public. Data conversion functions are provided. Data annotation follows a flexible schema (Grewe et al., 2011) so that any metadata necessary can be entered.

Recently, the INCF established the INCF Dataspace[9], a cloud based file system to federate all kinds of neuroscience data. There are several other initiatives (Marcus et al., 2007; Usui and Okumura, 2008; Moucek et al., 2014, etc.) that provide a web-based storage for different domains in neuroscience.

All these solutions are based on data exchange by files, and they provide little or no support for using formats or data structures that are in some way standardized. In most cases data are accessible only through a web browser. Interoperability between any of these solutions, or with other tools and formats used by neuroscientists, does not exist. As a basis for such interoperability, common standards for representing and accessing data would be needed, and tools and services to apply and use these standards also within the lab would have to be available. Such standardization will become also highly relevant for the recently initiated large-scale projects with strong electrophysiology components, such as the Allen Institute's Project Mindscope[10], the Human Brain Project[11], or the BRAIN Initiative[12].

[1] http://neurodatabase.org
[2] https://senselab.med.yale.edu
[3] http://www.neuinfo.org
[4] http://incf.org
[5] http://crcns.org
[6] http://www.carmen.org.uk/
[7] http://www.carmen.org.uk/standards
[8] https://portal.g-node.org/data/
[9] http://incf.org/dataspace
[10] http://www.frontiersin.org/10.3389/conf.fncom.2012.55.00033/event_abstract
[11] https://www.humanbrainproject.eu/
[12] http://www.nih.gov/science/brain/

Here we present GNData, the new version of the G-Node data management platform. This advanced version was developed with the aim not only to set up a repository of data files, but to provide a comprehensive framework that scientists can use to manage, access, and work with their data within their local laboratory workflow. The principal novelty of this framework is a standardized Application Programming Interface (API) together with client tools that enable data access directly from the local computational and/or laboratory environment. A unique feature is the ability to store and organize both the recorded data and the metadata together so that all information necessary for data analysis, re-analysis, and sharing is available in a unified way, and accessible through a well-defined interface. The integration of data and metadata has the benefits that data handling in the laboratory from recording to analysis becomes more efficient and reproducible, and that data sharing requires no further effort because all the information is already available with the data. Additionally, GNData allows for data sharing with colleagues, collaborators, or the public without any obstacles.

2. APPROACH

GNData addresses the need for comprehensive data management by providing (1) a storage system based on a common data representation that is suitable to organize data and metadata from any electrophysiological experiment, with (2) a general API so that data access can be integrated in software applications, and (3) client tools in common languages to support and facilitate this integration into the laboratory data workflow. These components implement a unique and efficient way of experimental data and metadata management, compared to the file-based systems.

2.1. REPRESENTATION OF DATA OBJECTS

A key element supporting reproducibility and data sharing is the standardization of formats and data structures. Using common data objects facilitates data access and data exchange, as well as the application of analysis tools. However, to be useful, standards must be applicable to the entire field without constraining the ability to store what is necessary. Given the variety and heterogeneity of electrophysiological studies, this poses the challenge of finding a balance between strict definitions to achieve the necessary standardization and flexible methods to account for the needs of any use case. GNData achieves this balance by combining a fixed data model for the recorded data with an adaptable and maximally unconstrained format for the metadata.

For the representation of electrophysiological data, the Neo python objects[13] are widely used (Garcia et al., 2014) and have come close to being a de-facto standard for describing recorded electrophysiological signals. Neo defines an object model with attributes and relationships that accounts for all types of recorded data (signal and spike data, multi-electrode data etc.), including numerical values, units, and dimensions. A typical Neo experimental representation is a dataset (named *Block* in Neo) containing several experimental trials (*Segments*), each having time series (*AnalogSignals*), spike event data (*SpikeTrains*) and stimulus event times as Neo *Events*. A dataset (*Block*) usually also contains groups of electrodes (*Recording Channel Groups*, *Recording Channels*) related with the recorded signals to indicate spatial position and arrangement of electrodes, and units (*Units*) identified by spike sorting as sources of spike trains (*SpikeTrains*).

In addition to the recorded data, GNData integrates metadata based on the open metadata Markup Language, odML[14] (Grewe et al., 2011). odML is an open, flexible and easy to use format to organize metadata in a hierarchical structure of key-value pairs (odML *Properties*). It provides a common *Section* object, which is used to meaningfully group *Properties* according to experimental aspects (Subject, Preparation, Stimulus, Hardware Settings etc.). *Section*s can be nested, enabling a flexible way to organize experimental metadata in a hierarchy that reflects the structure of the experiment. Thus, data annotation can be adapted to the requirements of each specific study. In addition, odML supports standardization by providing common terminologies[15]—pre-defined odML Section templates for typical experimental aspects to facilitate standardized descriptions of experiments across labs (Grewe et al., 2011).

Combining the Neo and odML concepts in a common object model, GNData integrates data and metadata in a unified framework. An example of the resulting data representation is illustrated in **Figure 2**.

2.2. COMMON INTERFACE TO ACCESS DATA AND METADATA

GNData integrates the Neo data model for electrophysiological data with the flexible odML data annotation under a single API definition. A common API is crucial as it unifies data management approaches, provides a defined way of data access, and makes data and metadata accessible to software tools. Previous approaches (Garcia et al., 2014, The Neuroshare Project[16]) have focused on representation of the recorded data. We complement these designs by integrating the essential methods for data annotation and permissions control, as well as providing a network-accessible implementation.

2.3. CLIENT LIBRARIES FOR MAIN COMPUTATIONAL PLATFORMS

The common data API of the GNData platform enables programmatic data access and data management through custom software. To support the use of the data API for everyday data management in the lab, we provide client libraries that communicate with the server via the GNData API, enabling instant data access from the local computational environment. Currently the focus is on Matlab and Python, which are among the most popular computational frameworks in experimental neuroscience. These client libraries[17] hide the generic API interface from the user and translate the commands and data to representations in the scientist's familiar environment, such as Neo Python objects (Garcia et al., 2014) in the Python client or Matlab structures in

[13] http://neuralensemble.org/neo/

[14] http://www.g-node.org/projects/odml

[15] http://www.g-node.org/projects/odml/terminologies

[16] http://neuroshare.sourceforge.net/API-Documentation/NeuroshareAPI-1-3.htm

[17] http://g-node.github.io/python-gnode-client, http://github.com/G-Node/gnode-client-matlab

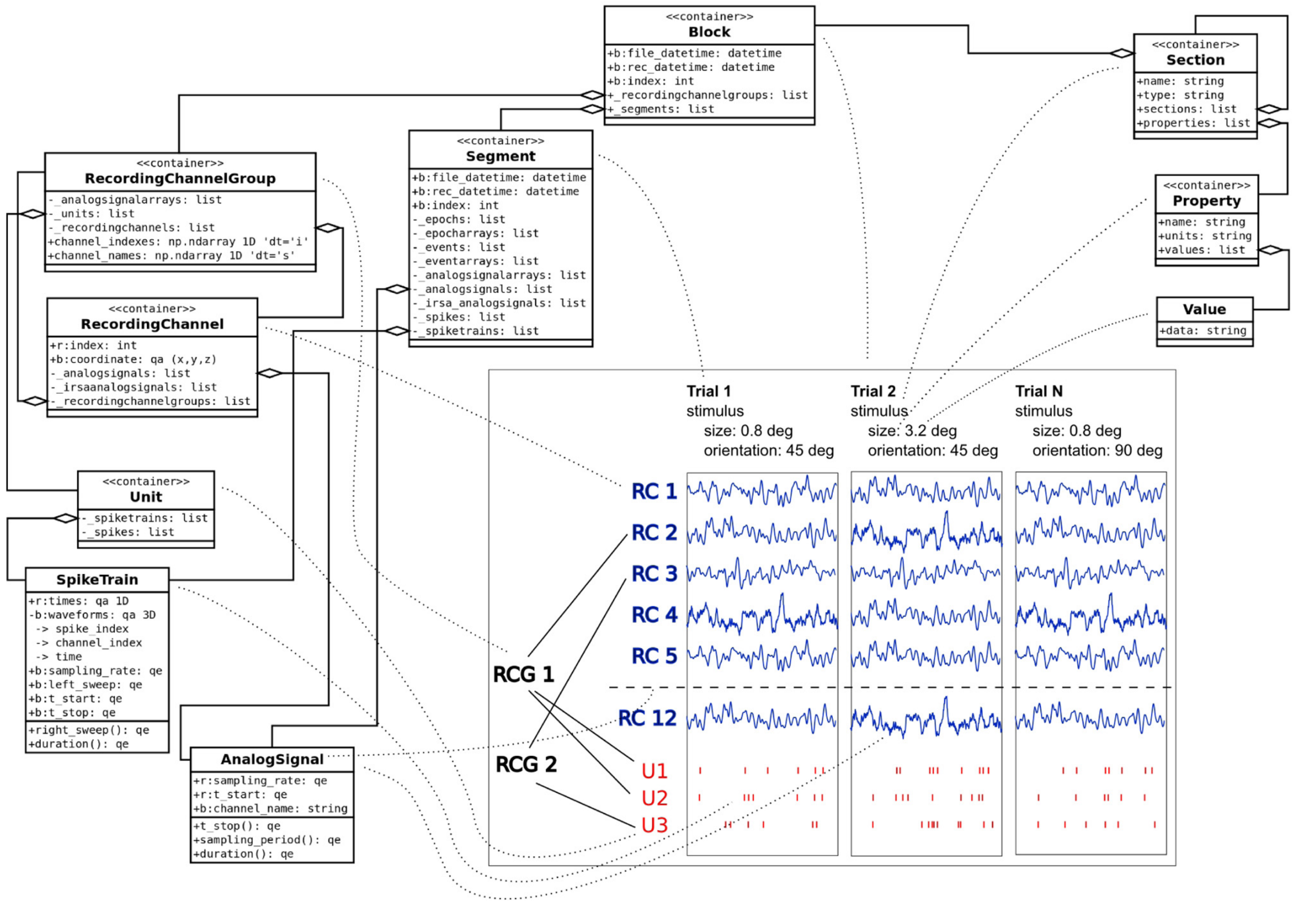


FIGURE 2 | Example data and metadata structure of an experiment with stimulation changing across trials. The panel on the bottom right represents the recorded signals in a study that investigates receptive fields of neurons in visual cortex of macaque monkeys. Each trial had its unique stimulus configuration (orientation, size, etc.). Local Field Potentials from different channels (RC1–RC12) were recorded during the experiment; spike trains of single units (U1–U3) were obtained by spike sorting. The dotted lines are used to represent a mapping between experimental entities and their representations in the object model. Neo objects[13] (left) are used to represent the data part of the experiment. odML[14] Section, Property and Value objects (top right) describe stimulus metadata, changing from trial to trial within a given experiment.

the Matlab client (see Appendix in Supplementary Material), thus enabling direct access to the data from the simulation software or analysis script. In addition, a web interface is provided for browser-based access. Enabling different types of data access supports interoperability and makes data access independent of a certain format, language, or platform.

2.4. DATA SHARING

As a multi-user system designed to facilitate collaborative research, GNData provides fine-grained mechanisms for access control and data sharing. Original data is always accessible for its owner. Any subset of data or metadata entities can be shared with selected users, for example collaborators. The ability to instantly access the same data without additional data transfer increases the efficiency of collaborative work. In addition, data can be opened to all users for public access. Thus, it is easy to provide data together with metadata for a data publication, to make selected data available for testing or benchmark purposes, or to release data to the public for re-use.

3. IMPLEMENTATION

This section describes the main implementation concepts of the GNData API. A full API reference can be found at the documentation page[18]. A demo environment is available where some of the examples below can be tested to get more detailed overview (Note that object identifiers can be different). Information about the demo environment is provided in the Appendix in Supplementary Material.

3.1. REST-FUL INTERFACE

The GNData API is built according to the REST principles (Fielding and Taylor, 2002). The REST protocol is designed for data representation supporting caching, scalability and client-server architecture[19]. Many stable open source libraries are available that support REST in different programming languages.

[18] http://g-node.github.io/g-node-portal/

[19] http://en.wikipedia.org/wiki/Client-server_model

One of the principles of REST is that every object has its permanent location defined via a Uniform Resource Locator (URL):

```
GET /electrophysiology/event/JD53GRU249/
```

In GNData , URLs defining object locations are designed to have a certain structure. The first part of the URL defines a namespace that corresponds to a particular neuroscientific domain or function. Currently, GNData supports “electrophysiology,” “metadata,” and “datafiles” namespaces, providing electrophysiological data, metadata, and file management functions, respectively. Every namespace includes a set of objects related to this particular namespace. The names of these object types form the second part of the URL. For instance, the “electrophysiology” namespace supports “event,” “spiketrain,” and other object types defined by Neo (see 2.1). The “metadata” namespace supports “section,” “property,” and “value” object types as provided by the odML definition. A unique base-32 (RFC4648) object identifier forms the end part of the object location.

Objects have consistent structured representations (see section 3.2) according to the integrated data model. For all objects, the GNData API supports a number of standard functions like creating, updating and deleting single objects, or making bulk object updates. A standard HTTP GET request selects one or several objects; requests are highly parameterizable, allowing filtering or processing objects in chunks:

```
GET /electrophysiology/event/?owner=demo&label=stimulus
```

In this example an HTTP GET request queries all event-type objects owned by the user “demo” having a label attribute equal to “stimulus.”

HTTP POST request type with JSON-encoded[20] data is used for making updates or creating new objects. HTTP DELETE is used to remove objects within the system; removed object will no longer appear in GET responses and will not be available for POST updates. However, removed objects are still accessible after the delete operation as all changes to object status and attributes are being tracked by the system (see section 4.4). Supported operations together with corresponding URL structures are listed in **Table 1**.

3.2. OPERATIONS WITH HTTP REQUEST AND RESPONSE

All operations use JSON[20] as a main request and response format. The JSON format is supported natively by Javascript[21] and also by many other common programming languages like Python [22] or Matlab [23].

As defined in **Table 1**, a GET request of the GNData API has the following form:

```
GET /<namespace>/<object_type>/[<object_id>]/[?<params>]
```

For example,

```
GET /electrophysiology/spiketrain/BE8O27N959/
```

returns the spiketrain object with the identifier “BE8O27N959” in JSON format. In order to create or update objects, a POST request with the same URL syntax is sent. For example,

```
POST /electrophysiology/spiketrain/BE8O27N959/
{
    "name": "SP-BE8O27N959",
    "comment": "spiketrain generated using wave_clus."
}
```

will make an update fields “name” and “comment” in the spiketrain object with identifier BE8O27N959.

A successful response contains the object represented in JSON format:

```
HTTP SUCCESS (200)

{
  "logged_in_as": "demo",
  "objects_selected": 1,
  "selected": [
    {
      "fields": {
        "id": 2,
        "guid": "88aa2089cfc73e9231c5518702222e5b8bb0d",
         "name": "V1 FIX signals, trial 1",
         "analogsignal_set": [
          "/electrophysiology/analogsignal/F8LD1EGINL",
          "/electrophysiology/analogsignal/LPTUBS44FG",
        ],
        "current_state": 10,
        "safety_level": 3,
        "owner": "/profiles/profile/5",
        "date_created": "2012-07-26 17:16:07",
```

[20] http://www.json.org/

[21] http://en.wikipedia.org/wiki/JavaScript

[22] http://docs.python.org/2/library/json.html

[23] http://www.mathworks.com/matlabcentral/fileexchange/20565-json-parser

Table 1 | Common API actions for every supported HTTP request type (GET, POST, DELETE) and typical request URL structure.

URL structure	Get	Post	Delete
/namespace/object_type/	List objects, apply filters	Create new object or make bulk update	Bulk delete
/namespace/object_type/id/	Access single object	Update single object	Delete single object
/namespace/object_type/id/acl/	Get object permissions	Update object permissions	405 Not Supported

Left column contains structure of the REST URL. Table cells define an action for each URL structure and HTTP request type.

```
        "block": "/electrophysiology/block/BDA1OIE4PE",
        "metadata": []
      },
      "model": "neo_api.segment",
      "permalink": "/electrophysiology/segment/RT4DALN1"
    },
    {
      ...
    }
    ],
  "selected_range": [ 0, 5 ],
  "message": "Here is the list of requested objects.",
  "message_type": "object_selected"
}
```

Exceptions are handled with standard HTTP response codes[24], such as 404—Object Not Found, 403—Forbidden, 400—Bad Request, 304 - Not Modified, etc., and the response body contains a JSON-formatted message with exception details.

3.3. DATA HANDLING

GNData uses the HDF5[25] file format in the backend to store array data. We made several performance tests against popular freely available data storage back-ends (PostgreSQL[26], MySQL[27] and HDF5; results not shown) resulting in HDF5 being an optimal solution for managing large data arrays, even with serial file access and fetching of multiple data slices. Every object in the GNData with associated array data (analog signal, spike train, waveform etc.) has the related data stored in HDF5 file. Data can be accessed by downloading a corresponding HDF5 file that contains an HDF5 array in the root of the file.

For data analysis, often only certain selected parts of the recorded data are desired. To reduce data transfer between client and server, a limited data slice can be requested. GNData supports partial data requests for objects with associated data array(s). This works for both single and multiple object requests and is practical when accessing large datasets.

The following request

```
GET /electrophysiology/analogsignal/LPTUBS44FG/?
        start_time=50&duration=100
```

returns data samples falling within the 100 ms time window of the originally recorded signal with ID = LPTUBS44FG starting from 50 ms (units are taken from object attributes). The response (not shown, see example in section 3.2) contains an URL to the corresponding file with a particular slice of array data,

```
GET /datafiles/8U1KHK8IA6/data/?
        start_index=1000&end_index=3001
```

This URL represents a link to a data file containing the data array for the selected analog signal, with parameters indicating the first and the last indexes of this array, needed to create the requested slice. These boundaries are calculated automatically based on the object attributes and their units (start time, sampling rate etc.) and request input parameters (start time, end time, duration etc.). Sending a GET request to this URL will download an HDF5 file containing raw data from the 100 ms time window only. Note that a datafile with array data is no longer dependent on the analog signal and will not contain units or other information, only data itself. All related meta information should be taken from the corresponding object.

3.4. CACHING

For efficiency, the GNData API is using standard HTTP mechanisms for data caching like e-Tags[28] and "last-modified" attributes in request headers. Every single change to an object results in a new e-Tag assigned to this object. By default, an object is not served for download if no changes were made and e-Tags of the previously downloaded object and the object on the server match. In this case, a standard 304 HTTP response ("Not Modified") is returned instead.

3.5. OPEN SOFTWARE AND MODULAR STRUCTURE

GNData is developed as an open source software based on the Django[29] framework. The framework is designed to be used with relational databases. Concurrent user access, as well as atomicity, consistency and isolation[30] are implemented on the database level. The software architecture follows a modular principle, so that implementation of new data models into the platform is straightforward[31]. The principal software components are illustrated in **Figure 3**.

The key programming language used is Python. Python has a growing community in Neuroscience with a large amount of open software available. The GNData project welcomes developers to contribute to the software[32].

4. KEY GNDATA FEATURES

In this section we describe key features and functional scope of the GNData platform.

4.1. DATA ACCESS WITH FILTERS

The GNData API provides query mechanism with different filters based on object attributes and relationships. It allows to query a subset of all available objects of a particular type based on certain criteria (equal, greater than, etc.), applied to object attributes. To avoid definition of a new query language, this query mechanism is built on top of the Django querying routine and uses similar concepts and namings[33]. Query parameters, specified in the request URL are directly converted into the request on the Django application level, with addition of certain authorization filters. The following examples illustrate the usage of filters.

This HTTP GET request will select metadata properties having "luminance" in their "name" attribute:

[24] http://www.w3.org/Protocols/HTTP/HTRESP.html
[25] http://www.hdfgroup.org/HDF5/
[26] http://www.postgresql.org/
[27] http://www.mysql.com/
[28] http://en.wikipedia.org/wiki/HTTP_ETag
[29] https://www.djangoproject.com/
[30] http://en.wikipedia.org/wiki/ACID
[31] https://docs.djangoproject.com/en/dev/topics/db/models/
[32] https://github.com/G-Node/g-node-portal/
[33] https://docs.djangoproject.com/en/dev/topics/db/queries/

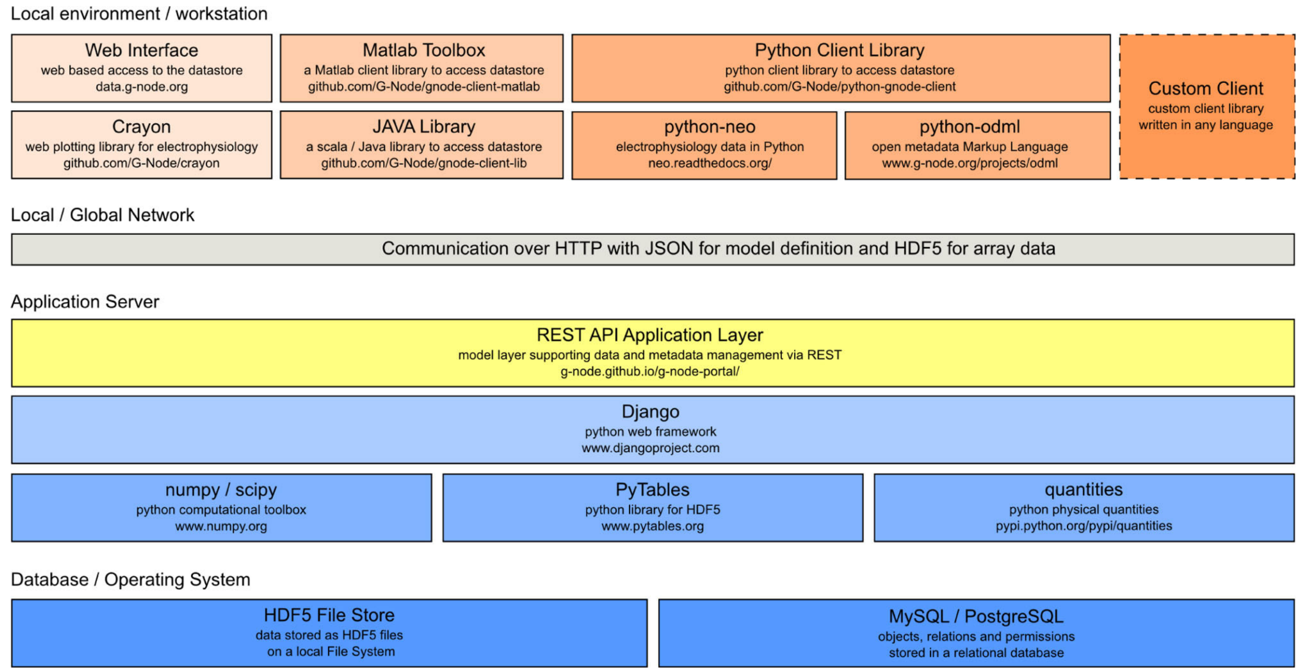

Local environment / workstation
Web Interface
web based access to the datastore
data.g-node.org
Matlab Toolbox
a Matlab client library to access datastore
github.com/G-Node/gnode-client-matlab
Python Client Library
python client library to access datastore
github.com/G-Node/python-gnode-client
Custom Client
custom client library
written in any language
Crayon
web plotting library for electrophysiology
github.com/G-Node/crayon
JAVA Library
a scala / Java library to access datastore
github.com/G-Node/gnode-client-lib
python-neo
electrophysiology data in Python
neo.readthedocs.org/
python-odml
open metadata Markup Language
www.g-node.org/projects/odml
Local / Global Network
Communication over HTTP with JSON for model definition and HDF5 for array data
Application Server
REST API Application Layer
model layer supporting data and metadata management via REST
g-node.github.io/g-node-portal/
Django
python web framework
www.djangoproject.com
numpy / scipy
python computational toolbox
www.numpy.org
PyTables
python library for HDF5
www.pytables.org
quantities
python physical quantities
pypi.python.org/pypi/quantities
Database / Operating System
HDF5 File Store
data stored as HDF5 files
on a local File System
MySQL / PostgreSQL
objects, relations and permissions
stored in a relational database

FIGURE 3 | GNData architecture diagram. From the bottom: integration of low-level data storage components (dark blue), application server components (light blue) with the REST API (yellow) as the common access interface. Top: Clients such as web interface components, Matlab client components, python client library, self-written custom clients.

```
GET /metadata/property/?name__icontains=luminance
```

The resulting response contains objects with their identifiers that can be used in another request for related objects (here "value" objects are actual values of related metadata properties):

```
GET /metadata/value/?
        parent_property__in=[AKDALFVCHL, PPQD09BPJ9]
```

Query conditions on related objects can be directly included in the request parameters. The next example request selects all metadata values of (related) properties with "luminance" in their "name" attribute:

```
GET /metadata/value/?
        parent_property__name__icontains=luminance
```

Filters allow to query for a certain subset of the experimental data, which can be used in analysis or visualization. **Figure 4** shows the plot of all LFP traces from a certain experimental trial, selected using filters with certain time and stimulus conditions. The query is explained in Appendix (Supplementary Material) in more detail.

A full query reference is available at the project documentation page.

4.2. UNIFIED ORGANIZATION OF DATA AND METADATA

GNData provides a common set of objects representing electrophysiological (experimental and/or simulated) data, together with an object model for flexible metadata description. The GNData API allows to establish meaningful connections between data and metadata objects. In particular, data objects can be hierarchically grouped (using odML *Sections*) to achieve an efficient organization. Any data object can be annotated by linking with the appropriate metadata objects, thus achieving a comprehensive data annotation for data selection and reproducibility. For example, consider an experiment where stimulus parameters change from trial to trial. In that case, for every experimental trial the appropriate stimulus property has to be indicated, which can be achieved by annotating each Neo *Segment* representing a trial to the appropriate metadata values.

Annotation is done by sending an HTTP POST request with the references to the metadata values and the target object for annotation:

```
POST /electrophysiology/analogsignal/F8LD1EGINL/
{
    "metadata": [
        "/metadata/values/KCAP5DK6FH/",
        "/metadata/values/JD53GRU249/"
    ]
}
```

In this example, an analog signal object with ID = F8LD1EGINL is annotated with certain metadata values (KCAP5DK6FH and JD53GRU249). Required values and their IDs can be pre-selected with another request using appropriate conditions and parameters (see section 4.1). These connections enable the researcher not only to identify the experimental context for a given data structure, but conversely also to query data by specific metadata. The

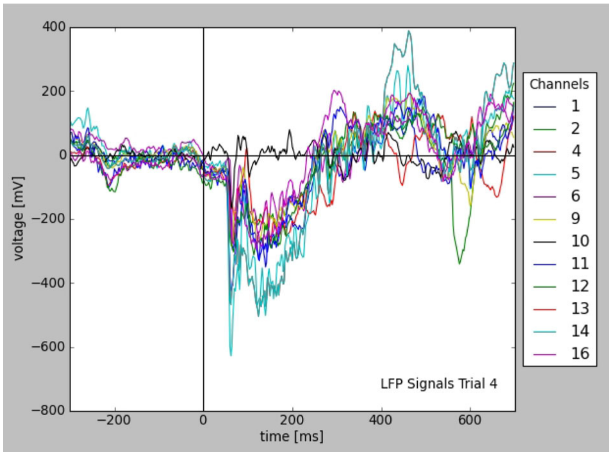


FIGURE 4 | Plot of LFP responses from a trial selected using certain time and stimulus conditions (see text). Note that all informations used for axes, labels, and legend were taken from the stored data and metadata directly.

following request selects all analog signal objects that have the above defined values as their metadata:

```
GET /electrophysiology/analogsignal/?
        ^1metadata=KCAP5DK6FH&^2metadata=JD53GRU249
```

In general, this allows using annotated metadata in requests for data object of any type.

4.3. DATA SHARING AND COLLABORATION

The GNData system provides a multi-user environment that facilitates collaborations where researchers need common access to datasets. Control of data access permissions is achieved using Access Control Lists (ACL), which provides several access levels for each object. By default, every object created in the system is private and only accessible for its owner. The owner of an object can make it accessible for an individual or group of collaborators, or open it for access to all users. Thus, data sharing can be done with a simple command, avoiding any data duplication or transfer.

GNData supports both read-only and read-write permissions for individual shares. The whole study can be opened for experimentalists for contribution with experimental recordings, while certain experimental trials can be made read-only for collaborators who perform data analysis.

Each object's current ACL is available for the object owner at a specific URL:

```
GET /<namespace>/<object_type>/<object_id>/acl/
```

The structure of the ACL in JSON format contains its global sharing level and the list of users having individual access:

```
HTTP SUCCESS (200)

{
    "safety_level": 1, # 1-private, 3-public
    "shared_with": {
       "userA": 1, # 1-read-only
       "userB", 2 # 2-read-write
    }
}
```

An authorized POST request to this URL with the request body containing new ACL configuration updates the object permissions.

4.4. VERSIONING

To support reproducibility, GNData implements object versioning mechanisms where all changes to any object are saved, and a user can always go back in time to the corresponding version of the data.

Requesting a certain version of an object is done by adding the "at_time" parameter to the GET request. The following example requests an object as it was at September, 15th 15:36:55:

```
GET /metadata/property/HB069BDMPG/?
        at_time=2013-11-09 15:36:55
```

5. DISCUSSION

We presented GNData, a data management system with an open API for electrophysiological data. GNData unifies organization of data and corresponding metadata and provides data access for researchers within a lab as well as for collaborators, directly from their computation environments. Efficient organization of data and metadata saves time for data access and facilitates data exchange and collaboration. Moreover, programmatic data access enables automatization of many steps in data collection and data organization, thus facilitating data analysis and collaborative research.

Key principle of the GNData architecture is an API that separates the user application from the storage backend and represents a consistent interface for accessing electrophysiological data. A common interface saves development and maintenance efforts and creates interoperability, faciliating application of tools and integration of software solutions. The GNData API combines a common representation for recorded data and a flexible metadata schema that is suitable to annotate data from any kind of experiment. This concept is independent of the REST implementation. Implementations in other programming languages and on different technologies are easily possible.

The GNData API includes basic functions for querying data using different filters applied to object attributes. More extensive search capabilities and support for complex queries would be desirable for data retrieval. Extended functionality based on existing open-source solutions (e.g., Lucene [34], Xapian [35], Minion[36], Elasticsearch[37] etc.) will be included in future releases.

The GNData platform provides a standardized data representation, but original recorded data can come from files in various formats. Using the G-Node Python client (Sobolev et al., 2014, see also Appendix in Supplementary Material) users can utilize the

[34] http://lucene.apache.org/

[35] http://xapian.org/

[36] https://minion.java.net/

[37] http://www.elasticsearch.org/

Neo I/O modules (Garcia et al., 2014) to read the data into Neo objects before uploading. However, ideally the central data server would provide this data conversion, so that users can upload their data in files which are automatically extracted to corresponding data structures on the platform. For this purpose, integration of the Neo I/O libraries into the GNData is under development.

Currently GNData is focused on electrophysiological data. Scalability of the data model and the Client-Server approach, however, allow straightforward extension to account for data from other fields of neuroscience. Neuromorphological data, imaging data or other types of data could be integrated simply by specifying the appropriate data models. The INCF Task Forces on Electrophysiology[38] and Neuroimaging[39] are currently working on standard data models and formats for the respective types of data (Teeters et al., 2013). Those standards will be integrated in the GNData platform as they are released. Likewise, to support the entire data processing workflow in the laboratory, results from data analysis need to be accommodated as well. This kind of extension will be introduced as one of the next steps.

GNData is developed as an open source project available at the public G-Node Github account[40]. The project is open to contribution from neuroscientists or members from other scientific fields.

FUNDING

Supported by the Federal Ministry of Education and Research (Grants 01GQ0801 and 01GQ1302).

ACKNOWLEDGMENTS

This work benefited from discussions with members of the INCF Program on Standards for Data Sharing task forces, of the CARMEN consortium, of the CRCNS initiative, and of the NeuralEnsemble initiative. We thank especially Raphael Ritz for discussions, feedback and contributions to the design of the platform, Andreas Herz for discussions, and Willi Schiegel and Tiziano Zito for technical support. We acknowledge the INCF Secretariat staff for providing a friendly environment for discussions and collaboration.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/journal/10.3389/fninf.2014.00032/abstract

REFERENCES

Austin, J., Jackson, T., Fletcher, M., Jessop, M., Liang, B., Weeks, M., et al. (2011). CARMEN: code analysis, repository and modeling for e-neuroscience. *Proc. Comput. Sci.* 4, 768–777. doi: 10.1016/j.procs.2011.04.081

Bandrowski, A. E., Bruha, P. P., Papez, V., Grewe, J., Moucek, R., Tripathy, S., et al. (2013). "Ontology for experimental neurophysiology: semantic annotations of neurophysiology data and metadata," in *2013 Society for Neuroscience Meeting* (San Diego, CA), Nov 9–13 2013.

Crasto, C. J., Marenco, L. N., Liu, N., Morse, T. M., Cheung, K. H., Lai, P. C., et al. (2007). SenseLab: new developments in disseminating neuroscience information. *Brief Bioinform.* 8, 150–162. doi: 10.1093/bib/bbm018

Fielding, R. T., and Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Trans. Internet Technol.* 2, 115–150. doi: 10.1145/514183.514185

Garcia, S., Guarino, D., Jaillet, F., Jennings, T. R., Pröpper, R., Rautenberg, P. L., et al. (2014). Neo: an object model for handling electrophysiology data in multiple formats. *Front. Neuroinform.* 8:10. doi: 10.3389/fninf.2014.00010

Gardner, D. (2004). Neurodatabase.org: networking the microelectrode. *Nat. Neurosci.* 7, 486–487. doi: 10.1038/nn0504-486

Gardner, D., Akil, H., Ascoli, G. A., Bowden, D. M., Bug, W., Donohue, D. E., et al. (2008). The neuroscience information framework: a data and knowledge environment for neuroscience. *Neuroinformatics* 6, 149–160. doi: 10.1007/s12021-008-9024-z

Gibson, F., Overton, P., Smulders, T., Schultz, S., Eglen, S., Ingram, C., et al. (2008). Minimum information about a neuroscience investigation (MINI): electrophysiology. *Nat. Precedings*. Available online at: http://precedings.nature.com/documents/1720/version/2

Grewe, J., Wachtler, T., and Benda, J. (2011). A bottom-up approach to data annotation in neurophysiology. *Front. Neuroinform.* 5:16. doi: 10.3389/fninf.2011.00016

Herz, A. V., Meier, R., Nawrot, M. P., Schiegel, W., and Zito, T. (2008). G-Node: an integrated tool-sharing platform to support cellular and systems neurophysiology in the age of global neuroinformatics. *Neural Netw.* 21, 1070–1075. doi: 10.1016/j.neunet.2008.05.011

Marcus, D. S., Olsen, T. R., Ramaratnam, M., and Buckner, R. L. (2007). The extensible neuroimaging archive toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34.

Moucek, R., Bruha, P., Jezek, P., Mautner, P., Novotny, J., Papez, V., et al. (2014). Software and hardware infrastructure for research in electrophysiology. *Front. Neuroinform.* 8:20. doi: 10.3389/fninf.2014.00020

Sobolev, A., Stoewer, A., Pereira, M., Kellner, C. J., Garbers, C., et al. (2014). Data management routines for reproducible research using the G-Node Python Client library. *Front. Neuroinform.* 8:15. doi: 10.3389/fninf.2014.00015

Teeters, J. L., Benda, J., Davison, A. P., Eglen, S., Gerhard, S., Gerkin, R. C., et al. (2013). Considerations for developing a standard for storing electrophysiology data in HDF5. *Front. Neuroinform. (Conference Abstract: Neuroinformatics 2013)* 69. doi: 10.3389/conf.fninf.2013.09.00069

Teeters, J. L., Harris, K. D., Millman, K. J., Olshausen, B. A., and Sommer, F. T. (2008). Data sharing for computational neuroscience. *Neuroinformatics* 6, 47–55. doi: 10.1007/s12021-008-9009-y

Usui, S., and Okumura, Y. (2008). "Basic scheme of neuroinformatics platform: XooNIps," in *Proceedings of the 2008 IEEE World Conference on Computational Intelligence: Research Frontiers* (Berlin; Heidelberg: Springer-Verlag), 102–116. Available online at: http://dl.acm.org/citation.cfm?id=1788915.1788921

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 24 December 2013; accepted: 17 March 2014; published online: 23 April 2014.

*Citation: Sobolev A, Stoewer A, Leonhardt A, Rautenberg PL, Kellner CJ, Garbers C and Wachtler T (2014) Integrated platform and API for electrophysiological data. Front. Neuroinform. **8**:32. doi: 10.3389/fninf.2014.00032*

This article was submitted to the journal Frontiers in Neuroinformatics.

[38] http://www.incf.org/programs/datasharing/electrophysiology-task-force

[39] http://www.incf.org/programs/datasharing/neuroimaging-task-force

[40] https://github.com/G-node

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 07 March 2014
doi: 10.3389/fninf.2014.00020

Software and hardware infrastructure for research in electrophysiology

***Roman Mouček*[1,2]*, *Petr Ježek*[1,2], *Lukáš Vařeka*[1], *Tomáš Řondík*[1,2], *Petr Brůha*[1,2], *Václav Papež*[1], *Pavel Mautner*[1], *Jiří Novotný*[1], *Tomáš Prokop*[1] and *Jan Štěbeták*[1]**

[1] *Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Plzeň, Czech Republic*
[2] *New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Plzeň, Czech Republic*

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Sam Neymotin, State University of New York, USA
Juliana A. Knocikova, Centre National de la Recherche Scientifique, France

****Correspondence:***
Roman Mouček, Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, Plzeň 306 14, Czech Republic
e-mail: moucek@kiv.zcu.cz

As in other areas of experimental science, operation of electrophysiological laboratory, design and performance of electrophysiological experiments, collection, storage and sharing of experimental data and metadata, analysis and interpretation of these data, and publication of results are time consuming activities. If these activities are well organized and supported by a suitable infrastructure, work efficiency of researchers increases significantly. This article deals with the main concepts, design, and development of software and hardware infrastructure for research in electrophysiology. The described infrastructure has been primarily developed for the needs of neuroinformatics laboratory at the University of West Bohemia, the Czech Republic. However, from the beginning it has been also designed and developed to be open and applicable in laboratories that do similar research. After introducing the laboratory and the whole architectural concept the individual parts of the infrastructure are described. The central element of the software infrastructure is a web-based portal that enables community researchers to store, share, download and search data and metadata from electrophysiological experiments. The data model, domain ontology and usage of semantic web languages and technologies are described. Current data publication policy used in the portal is briefly introduced. The registration of the portal within Neuroscience Information Framework is described. Then the methods used for processing of electrophysiological signals are presented. The specific modifications of these methods introduced by laboratory researches are summarized; the methods are organized into a laboratory workflow. Other parts of the software infrastructure include mobile and offline solutions for data/metadata storing and a hardware stimulator communicating with an EEG amplifier and recording software.

Keywords: electrophysiology, event related potentials, infrastructure, neuroinformatics, workflow, portal, signal processing methods, stimulator

1. INTRODUCTION

As in other areas of experimental science, operation of electrophysiological laboratory, design and performance of electrophysiological experiments, collection, storage and sharing of experimental data and metadata, analysis and interpretation of these data, and publication of results are time consuming activities. If these activities are well organized and supported by a suitable infrastructure, work efficiency of researchers increases significantly.

Our research group, a member of the Czech National Node of International Neuroinformatics Coordinating Facility (INCF, 2013), focuses on research of brain electrical activity using the methods and techniques of electroencephalography (EEG) and event related potentials (ERP). Our neuroinformatics laboratory, which started to operate in 2005, is currently equipped with a number of commercial and custom hardware devices and software tools. Besides the basic electrophysiological infrastructure (amplifier, synchronization device, recording and analytic software, and software for presentation of stimuli) the laboratory equipment includes a sound and electrically shielded booth, a car simulator including a car cockpit, wheel and pedals connected to the computer, projector, and software tools for the simulation of driving environment and driving itself. Since the group has been solving difficulties with the laboratory operation (software and hardware tools and the whole infrastructure) from the beginning of its research activities, this paper introduces not only the current state of the laboratory infrastructure but also some essential intermediate steps in its building. The presented infrastructure is also more oriented to the processing of data from ERP than EEG experiments; as a result e.g., the methods for ERP component detection are highlighted in the text.

The paper is organized in the following way. The section Materials and Methods contains the description of the state of the art in building infrastructures for research in electrophysiology and neurophysiology. The next subsections first introduce the whole concept of the laboratory infrastructure; then some infrastructural parts are described. The section Results provides information about the current state and some implementation details of the selected parts of the infrastructure. The section Discussion mainly discusses the potential limitations of the built infrastructure and primarily speculates on the future direction of the proposed infrastructure.

2. MATERIALS AND METHODS

2.1. STATE OF THE ART

Building large infrastructures for research has become very popular with the rapid development of computers and hardware devices, programming languages and technologies, software tools, and online communication. This development has also spread to neuroscience and neuroinformatics to support the efficiency of the research in the field.

Coordinating activities in neuroinformatics are led by INCF that develops and maintains database and computational infrastructure for neuroscientists. Software tools and standards for the international neuroinformatics community are being developed through the INCF Programs, which address infrastructural issues of high importance to the neuroscience community (INCF, 2013). To enable collaboration between researchers through the sharing of neuroscience data, INCF introduced the INCF Dataspace (INCF Group, 2013). It associates INCF nodes data sources in a distributed system based on iRods solution. Technically, data are managed locally by individual nodes and connected using catalog servers. From a user perspective it works as a large data file system accessed through a web interface. In other words, all these zone servers, connected resources, and hosted datasets build a distributed network of shared data.

In electrophysiology (and in neurophysiology in broader sense) we can also find activities focusing on building larger software and/or hardware infrastructures. These activities, carried out by universities, research institutions and private companies, include cooperation of various hardware devices supported by related software tools in laboratories, definition of data formats, solutions for storing, managing, and sharing data and metadata, and development of methods and workflows for data processing, visualization and interpretation. Finally, more complex, usually web based solutions then can serve as virtual laboratories. The following parts of this section introduce some of the various approaches and activities that contribute to building infrastructures in electrophysiology (neurophysiology). More complex and already existing infrastructures are also mentioned.

The description of the electrophysiological domain (and description of any domain in general) could be provided at different levels of abstraction and includes both cooperating and competing techniques and approaches (e.g., classical data modeling vs. ontological modeling). Moreover, various physical repositories are used to store domain data and metadata. Then various programing languages, coding and architectural styles, technologies, and software tools are used to process these data and metadata. Since it is out of scope of this paper to focus on and describe the differences and relationships between various techniques and approaches, the following selection just introduces well known approaches and activities.

Open Metadata Markup Language (odML) (Grewe et al., 2011) is a flexible and unified metadata format for annotation data in neurophysiology. This language defines terminologies for the domain, but simultaneously is extensible and flexible for science that continually changes, and does not restrict the user by requiring entries. It increases its potential to become an exchange/sharing format for electrophysiology data. Metadata stored in odML are linked to the related data, for which a suitable exchange/sharing format is also looked for. Currently, great deal of attention is paid to HDF5 (Hierarchical Data Format) (HDF Group, 2013), or similar formats based on HDF5 (e.g., epHDF). HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of data types, and is designed for flexible and efficient I/O and for high volume and complex data. NoSQL document databases, due to their flexibility, are also very promising for long term storage of electrophysiological data and metadata. We tested StorageBIT (Carreiras et al., 2013) that combines HDF5 and MongoDB. HDF5 ensures data persistency while MongoDB is a front end for data access. Our tests (Jezek et al., 2014) proves that MongoDB is equivalent to relational databases from the performance point of view. Moreover it provides a better flexibility.

One of the leading initiatives for data sharing is the Neuroscience Information Framework (NIF, 2013) as a dynamic inventory of registered web-based neuroscience resources (data, materials, and tools). NIF enables access to public research data and tools through an open source environment (Gardner et al., 2008). Currently, it is with more than 6,400 resources one of the largest collection of neuroscience data. Each new resource has to pass at least one of three levels of registration. These levels specify depth of resource integration into NIF; level 1 provides information about the resource, level 2 provides direct access to resource's web services, and finally, the resource is sustained by an ontology at level 3. G-Node data management platform is a sharing facility that allows data organization, annotation, access and sharing. All can be managed via web-based interface as well as via RESTful API; access using an external application is possible. G-Node provides Matlab and Python scripts clients (G-Node, 2013).

In addition to a proper data and metadata format, ontologies are also helpful for data sharing. Significant representatives of bio-ontologies dealing with neurophysiology and electrophysiology are Neural Electro Magnetic Ontologies NEMO (Dou et al., 2007) and the Ontology for Biomedical Investigations OBI (Brinkman et al., 2010). The ontology built within the NEMO project provides formal semantic definitions of concepts in ERP research, including ERP patterns, spatial, temporal, functional (cognitive/behavioral) attributes of these patterns, data acquisition and analysis methods (Dou et al., 2007). OBI is an ontology for biological and clinical investigation description. Its terminology contains domain-specific terms and universal terms for general biological and technical usage. Finally, the ontology will represent the design of an investigation, the protocols and instrumentation used, the material used, the data generated and the type analysis performed on it (Brinkman et al., 2010).

Methods, techniques and tools for ERP signal processing are also a very important part of the software infrastructure in electrophysiology. The standard approach for event-related potential (ERP) signal processing can be divided into following steps: analog to digital conversion, filtering, segmentation, latency correction, averaging, and methods for detection and analysis of ERP components. As a result, a set of parameters describing

ERP components is obtained. From this result, useful information about the medical condition of the measured subject can be determined. (Picton et al., 1995; Luck, 2005). Proven techniques for ERP signal processing include wavelet transform (Quiroga and Garcia, 2003), matching pursuit (Aviyente et al., 2006), Independent Component Analysis (ICA) (Makeig et al., 1997), Principal Component Analysis (PCA) (Dien, 2012), and Hilbert-Huang transform (HHT) (Cong et al., 2009). For subsequent classification, Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and multi-layer perceptron are among the most frequently used methods (Lotte et al., 2007).

Different signal processing tools have been used for event-related data potential data processing within neuroinformatics community. Matlab (MATLAB, 2012) is the most popular since it is easy to use and implements many signal processing methods—either in its core (e.g., temporal filtering, FFT), or in default toolboxes (wavelet transform, matching pursuit, etc.). Furthermore, EEGLAB (Delorme and Makeig, 2004) can be directly used for the analysis of EEG/ERP experiments. EEGLAB is an interactive Matlab toolbox for continuous and event-related neurophysiological data processing. It allows researchers to load data in various formats, to extract epochs using stimuli markers, to remove artifacts (e.g., by using ICA), etc. The BrainVision Analyzer (BrainProducts, 2012) is a complex tool for neurophysiological data analysis. It provides an easy-to-use user interface, multiple import and export features, different views for visualization, and methods for signal processing and analysis. EEGVIS (Robbins, 2012) is a MATLAB toolbox that allows users to quickly explore multi-channel EEG and other large array-based data sets using multi-scale drill-down techniques. This toolbox can be used directly in MATLAB at any stage in a processing pipeline, as a plug-in for EEGLAB, or as a standalone precompiled application without MATLAB running.

The CARMEN project (Carmen, 2013) is an effort to create a virtual laboratory. It allows neuroscientists to share and exploit data, programs (services) and expertise from neurophysiological experiments. Neural activity recordings (signals and image series) are the primary data types. The CARMEN Portal is a web interface onto the CARMEN system accessed via a standard web browser that provides users with access to the computer and data storage resources. The project also developed a workflow generation and execution system within the platform. The Java-based CARMEN Workflow Tool consists of a graphical design tool, a workflow engine, and access to a library of CARMEN services and common workflow tasks. It supports both data and control flow, and allows parallel execution of services (Carmen, 2013).

Several initiatives and/or pilot studies also try to provide a solution for researchers to efficiently work out of laboratories using portable devices as laptops, tablets or mobile phones. Clinician Assessment and Remote Administration Tablet (CARAT) (Turner et al., 2011) is a Microsoft Windows tablet adapted to collect and administer clinical assessments in large scale demographic or neuropsychiatric studies. It uses an architecture with two modules. The first one set-ups the clinical study while the second one serves to data collection. Collected data are synchronized with a remote database. Research Electronic Data Capture (Harris, 2012) (REDCap) is a software application and workflow methodology designed to collect and manage data for research studies. REDCap Mobile (Borlawsky et al., 2011) is a solution that describes encrypted laptops with a push-pull relationship to the centralized REDCap database to allow data collection while off-line. Such solution is suitable in studies that need to be performed on places without an internet access as hospitals or jails.

Devices for presenting and synchronizing stimuli and responses to them are also an important part of the infrastructure in electrophysiology. Hardware and software stimulators are produced and sold by multinational companies Metrovision, LKC Technologies, Grass Technologies, Inomed and Neurobehavioral systems. Their production is usually very specific and intended for medical purposes.

2.2. OVERALL ARCHITECTURE

The overall architecture of the software and hardware infrastructure for research in electrophysiology comes from the set of the main activities performed by researchers during electrophysiological experiments. First, hypotheses and design of protocols for specific experiments are done. Then experiments are performed according to defined scenarios (protocols) and data and related metadata are collected. During the experiment the EEG signal obtained from the scalp of the tested subject is synchronized with presented stimuli. Second, the data are analyzed using various processing methods. Then, the data are interpreted and the results are published. The biggest obstacles for science are the following: since data are not well-described, conclusions and interpretations cannot be later reproduced or verified. The methods used for data analysis are lost or their detailed parameters are not later traceable. To solve these difficulties, initiatives as (Teeters et al., 2008) have been established to support experimental data sharing. The development of the complex infrastructure for experiments in the EEG/ERP domain contributes to international efforts in the electrophysiology domain. An overall architecture of this infrastructure is shown in **Figure 1**.

The basic aim of this infrastructure is to increase effectiveness and efficiency of scientific research in the field. The central point of the infrastructure, the EEG/ERP Portal (EEG/ERP Portal, 2013), is a service providing interface to human users and software tools. The main features of this service include long-term and sustainable storage of data and related metadata collected from experiments, various methods and workflows for data processing, and sharing of data, documents, methods and workflows in groups.

An initial idea of this infrastructure was particularly described in Jezek et al. (2013b). Besides a classical web based interface intended for human readers several communication interfaces for external tools have been implemented. Standalone tools including JERPA, offline and mobile version of the EEG/ERP Portal, or tools for signals visualization communicate with the EEG/ERP Portal using web services. Other tools as a Semantic Framework are implemented as libraries integrated directly within the EEG/ERP Portal. A substantial part of a complex infrastructure is created by several third-party hardware devices and software tools. These devices and tools are controlled by the experimenter who interacts

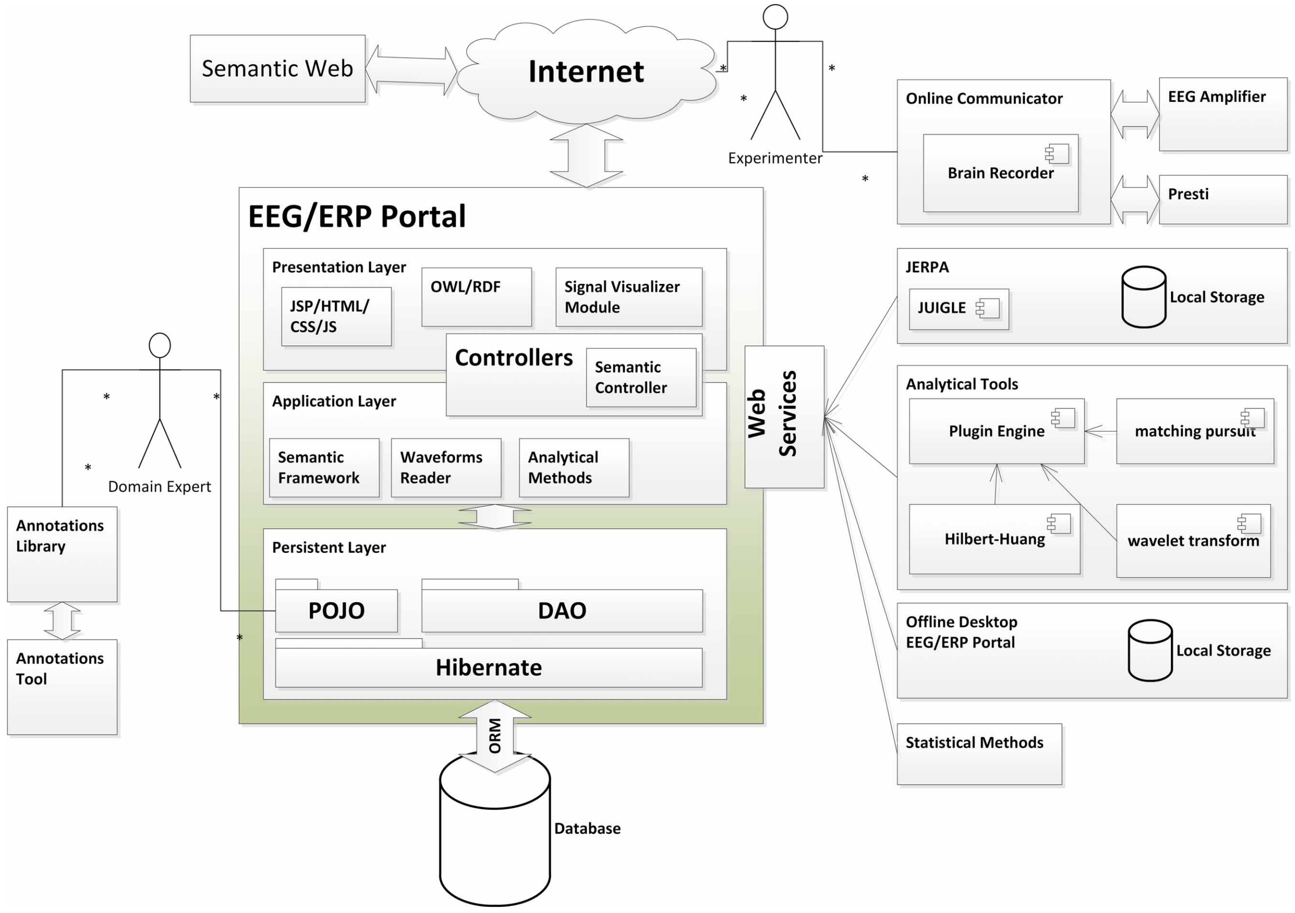


FIGURE 1 | Overall architecture (Jezek et al., 2013b). *Means "many" (0-n occurrences) relationship.

with the EEG/ERP Portal using a web browser on a standard computer, or using a mobile version of the EEG/ERP Portal.

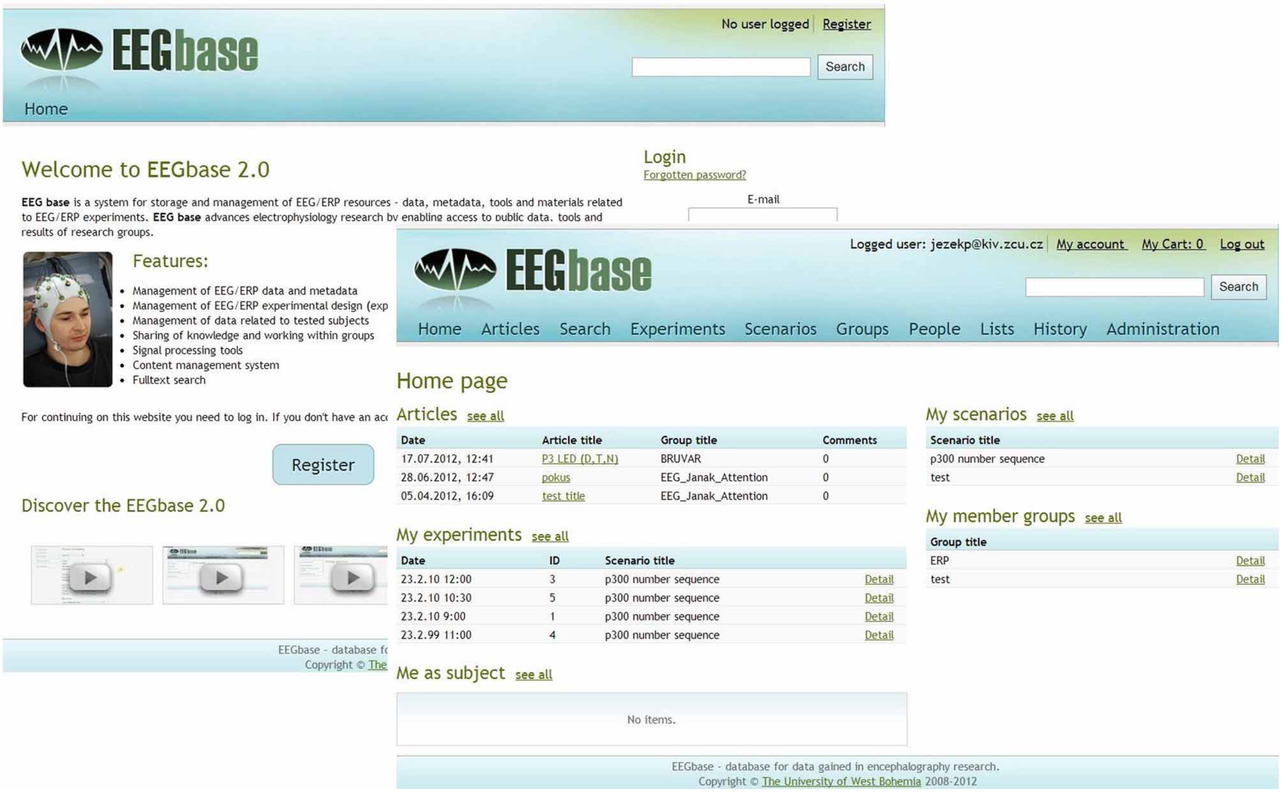
2.3. EEG/ERP PORTAL

The EEG/ERP Portal is a mature web-based system that enables researchers to upload, download and manage EEG/ERP experiments (data, metadata, experimental scenarios, etc.) (Jezek and Moucek, 2012b). The features of the EEG/ERP Portal also include sharing of knowledge, working in research groups, manage scientific discussions, run methods for signal processing, etc.

Different users have different roles in the system and the related level of authority. The users' credentials are required when users access the system. Individual users are grouped into self-managed groups. The user who wants to upload or download experiments has to be registered within the system and has to become a member of at least one group. On the basis of activities that the user can perform, several user roles are defined (Reader, Experimenter, Group Administrator, and Supervisor).

A simple wizard that guides the logged user through the process of adding an experiment facilitates upload of an experiment. Each experiment contains raw data supplemented by related metadata. A set of metadata which the user is instructed to fill in through the prepared forms is defined. These metadata are organized in semantic groups (experimental protocol, experimenters and tested subjects, used hardware, description of raw data, etc.) in accordance with an internal ontology initially presented in Jezek and Moucek (2011b). The experimenter can also decide if the experiment is private or public. Public experiments are downloadable for all registered users (without personal data of tested subjects), while private experiments are downloadable only within the experimenter's group. The functionality that includes possibility to associate experiments into experimental packages is in development. Individual packages can have a different access level. Experiments in these packages can be managed in bulk. The overall preview of the EEG/ERP Portal is shown in **Figure 2**.

Since the EEG/ERP Portal cooperates with a set of associated submodules, several communication interfaces for external tools have been designed and implemented. The tools can be divided into two groups. The first group includes tools accessible through an internet browser. These tools are implemented as stand-alone libraries integrated within the EEG/ERP Portal directly. The most important tool is the Semantic Framework (Jezek and Moucek, 2012a). The aim of the Semantic Framework is to provide experimental metadata in the semantic web languages and technologies (RDF, OWL). Data expressed in these languages and technologies are readable by semantic reasoners.


EEGbase
No user logged | Register
Search
Home
Login
Forgotten password?
E-mail
Register
Discover the EEGbase 2.0
Logged user: jezekp@kiv.zcu.cz | My account My Cart: 0 Log out
Home Articles Search Experiments Scenarios Groups People Lists History Administration

Welcome to EEGbase 2.0

EEG base is a system for storage and management of EEG/ERP resources - data, metadata, tools and materials related to EEG/ERP experiments. **EEG base** advances electrophysiology research bv enabling access to public data. tools and results of research groups.

Features:

- Management of EEG/ERP data and metadata
- Management of EEG/ERP experimental design (exp
- Management of data related to tested subjects
- Sharing of knowledge and working within groups
- Signal processing tools
- Content management system
- Fulltext search

For continuing on this website you need to log in. If you don't have an ac

EEGbase - database fo
Copyright © The

Home page

Articles see all

Date	Article title	Group title	Comments
17.07.2012, 12:41	P3 LED (D,T,N)	BRUVAR	0
28.06.2012, 12:47	pokus	EEG_Janak_Attention	0
05.04.2012, 16:09	test title	EEG_Janak_Attention	0

My experiments see all

Date	ID	Scenario title	
23.2.10 12:00	3	p300 number sequence	Detail
23.2.10 10:30	5	p300 number sequence	Detail
23.2.10 9:00	1	p300 number sequence	Detail
23.2.99 11:00	4	p300 number sequence	Detail

Me as subject see all

No items.

My scenarios see all

Scenario title	
p300 number sequence	Detail
test	Detail

My member groups see all

Group title	
ERP	Detail
test	Detail

EEGbase - database for data gained in encephalography research.
Copyright © The University of West Bohemia 2008-2012

FIGURE 2 | EEG/ERP Portal Overview. The login page and the home page of a logged user are shown. The logged user can see summarized information about his/her activities.

The second group of tools includes desktop or web-based tools that run locally on the user's computer. These tools access data in the EEG/ERP Portal and these data are then processed locally. The Electroencephalography Data Processor (EEG Data Processor) (Jezek and Moucek, 2013b) is a system for running methods for signal processing that enables a remote processing of data from the EEG/ERP Portal. The methods for signal processing are installed as plug-ins. Data can be uploaded directly using the web interface, or through the web service endpoint.

2.4. DATA MODELS AND ONTOLOGIES

The data model of the EEG/ERP Portal was first proposed in 2008 and since then has changed several times. Currently the core ERA model contains more than 70 tables. However, the flexibility of the data model and possibility to share data within community are still more important in recent years. Then, the main goal of data model improvement and ontology development is to increase data sharing abilities of the Portal. Currently, ontologies have become not only recommended but even required domain descriptions (e.g., NIF third level registration requires an ontological description). Besides existing projects a new Ontology for describing Experimental Neurophysiology (OEN) (Bruha et al., 2013) is being developed. The group working on the development of this ontology was formed from the members of the following initiatives:

- EEG/ERP Portal (EEGBase) (http://eegdatabase.kiv.zcu.cz/home.html)
- G-Node (www.g-node.org)
- INCF Task Force on standards for sharing of electrophysiology data (http://www.incf.org/programs/datasharing/electrophysiology-task-force)
- NIF (www.neuinfo.org)
- Neuroelectro.org (www.neurolectro.org)

The group follows the best practices for creating ontologies, for example, it cooperates with community of researchers who design and create ontologies, uses existing data formats and repositories (odML, HDF5), and reuses existing resources (terms, ontologies - NEMO, OBI). For the general description of experimental neurophysiology, the terms from ontologies NEMO and OBI are relevant. However, the set of the domain terms is still not complete in these ontologies (information stored in the EEG/ERP Portal cannot be fully described by these ontologies) and OEN will be finally an extension of OBI (e.g., the granularity of OBI for devices and related information will be extended).

Currently, the development of OEN has been separated into two branches. The first branch deals with structured terminology to annotate experimental metadata (e.g., devices or methods); the second branch deals with structured terminology to annotate experimental data (e.g., action potential). The knowledge model of 'device branch' is shown in **Figure 3**.

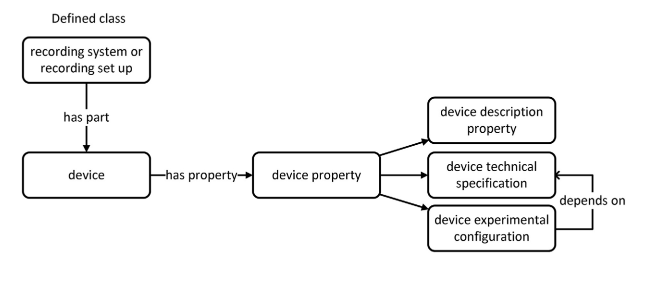

Defined class
recording system or recording set up
has part
device
has property
device property
device description property
device technical specification
device experimental configuration
depends on

FIGURE 3 | Device knowledge model (Bruha et al., 2013).

Terminologies within OEN have been primarily developed in the odML format. Subsequently, an OWL file has been constructed aided by Ontofox (Xiang et al., 2010). The current developer's version of OEN is available at https://github.com/G-Node/OEN.

To define the terms and to create the 'device branch' of OEN the following schemas that describe the elements of experimental setups and interactions between these elements were used: the experimental setup for investigation of driver's attention (**Figure 4**), experimental setup for performing traditional oddball experiment, and experimental setup for investigation of mouse visual cortex. Based on these schemas the preliminary knowledge model representing the experimental setup has been constructed. This model can be used to annotate the EEG/ERP Portal.

2.5. SIGNAL PROCESSING METHODS

A subset of signal processing methods suitable for ERP waveforms detection and classification, alternatively for clustering the feature vectors extracted from ERP signal was investigated, modified and implemented by the members of our research group. These methods can be run as web services within the Electroencephalography Data Processor.

Hilbert-Huang transform [HHT, see (Huang et al., 1998) for details] is a signal processing method designed especially for non-linear non-stationary signals. It consists of empirical mode decomposition (EMD) and Hilbert spectral analysis (HSA). During the process called sifting EMD decomposes signal to intrinsic mode functions (IMF) and residue. HSA computes an analytical signal from IMF and then analytical signal instantaneous attributes. Original HHT algorithm is not fully suitable for EEG signal processing, because EEG is a quasi-stationary signal. In Ciniburk (2011) we introduced the way HHT can be used for ERP waveforms detection. In Prokop (2013) we introduced particular improvements of the classifiers for ERP waveform detection that work with HHT results. Currently, the classification reliability of the ERP detection by the modified HHT is comparable with continuous wavelet transform and matching pursuit algorithm (see Ciniburk, 2011).

The traditional matching pursuit algorithm (MP) as proposed by Mallat and Zhang (1993) is suitable for EEG/ERP signal processing because the subset of atoms from the Gabor base is correlated with ERP components Benar et al. (2007). However, the computational complexity of its brute force implementation is challenging for on-line calculations. One of the most promising implementations (Ferrando et al., 2002) is based on restricting the combinations of Gabor parameters that need to be used for scalar product calculations, an approximation of the original signal. We showed that MP with GD can be used as a suitable preprocessing method for the task of ERP detection based on a classifier which works with Vigner-Wille transform of MP result. We identified a few issues which led to false positive/negative ERP waveform detection results and solved some of them. In Rondik (2010) we based the classification on correlation between a model of ERP waveform and a signal reconstructed from significant atoms. We also introduced solution for the case if an ERP component is approximated by two or more atoms.

Various algorithms were investigated regarding their benefits for off-line BCI systems, and ERP component detection. For the P300 BCIs purposes, a multi-layer perceptron (MLP) was used to classify the features obtained using matching pursuit (Vareka, 2012) and discrete wavelet transform (Vareka and Mautner, 2013). For the MLP design, one hidden layer was used and the number of neurons was optimized using a validation dataset. The main goal of the research was to evaluate if the multilayer perceptron is suitable for the P300 detection and to find the architectures and training algorithms that perform comparably well for this task. We were able to prove on the off-line dataset that the trained MLP neural network with the architecture described in Vareka (2012) is able to detect the P300 component as successfully as other state-of-the-art classification approaches.

Neural networks have also been used to cluster the feature vectors that were extracted from the ERP signal. The ERP signal reflects not only ERP components, but also artifacts and background EEG activity. The objective was to analyze the signal and to try to separate different waveforms without using reliable but computationally complex Independent Component Analysis as proposed in Makeig et al. (1997). Furthermore, since the latency of ERP components may vary for different subjects, or stimulation protocols (Luck, 2005), the method can also be used to cluster the feature vectors assigned to a specific ERP component to further analyze how the component might be affected by external factors or disease. In Vareka and Mautner (2012), the features were extracted from the signal using matching pursuit (Mallat and Zhang, 1993). The ART 2 neural network (Carpenter and Grossberg, 1987) was used to cluster the ERP features. The optimal adjusting parameters for the ART 2 neural network were found. As a result, the traditional ART 2 network was proven to be useful for our experiments. The proposed architecture is described in more detail in Vareka and Mautner (2012).

2.6. WORKFLOWS

Data obtained from electrophysiological experiments are analyzed using various preprocessing and processing methods, some of them are described in Section 2.5. However, there is usually a need to use more than one method for analysis of the EEG/ERP signal. Therefore, we provide an opportunity to define workflows for complex analysis of experimental data. In our domain, a workflow includes a complex set of analytic methods that process experimental data sequentially or in parallel. It is organized as a tree structure, where each branch of the tree has the same meaning as a pipe in Linux; an output of the method serves as an input

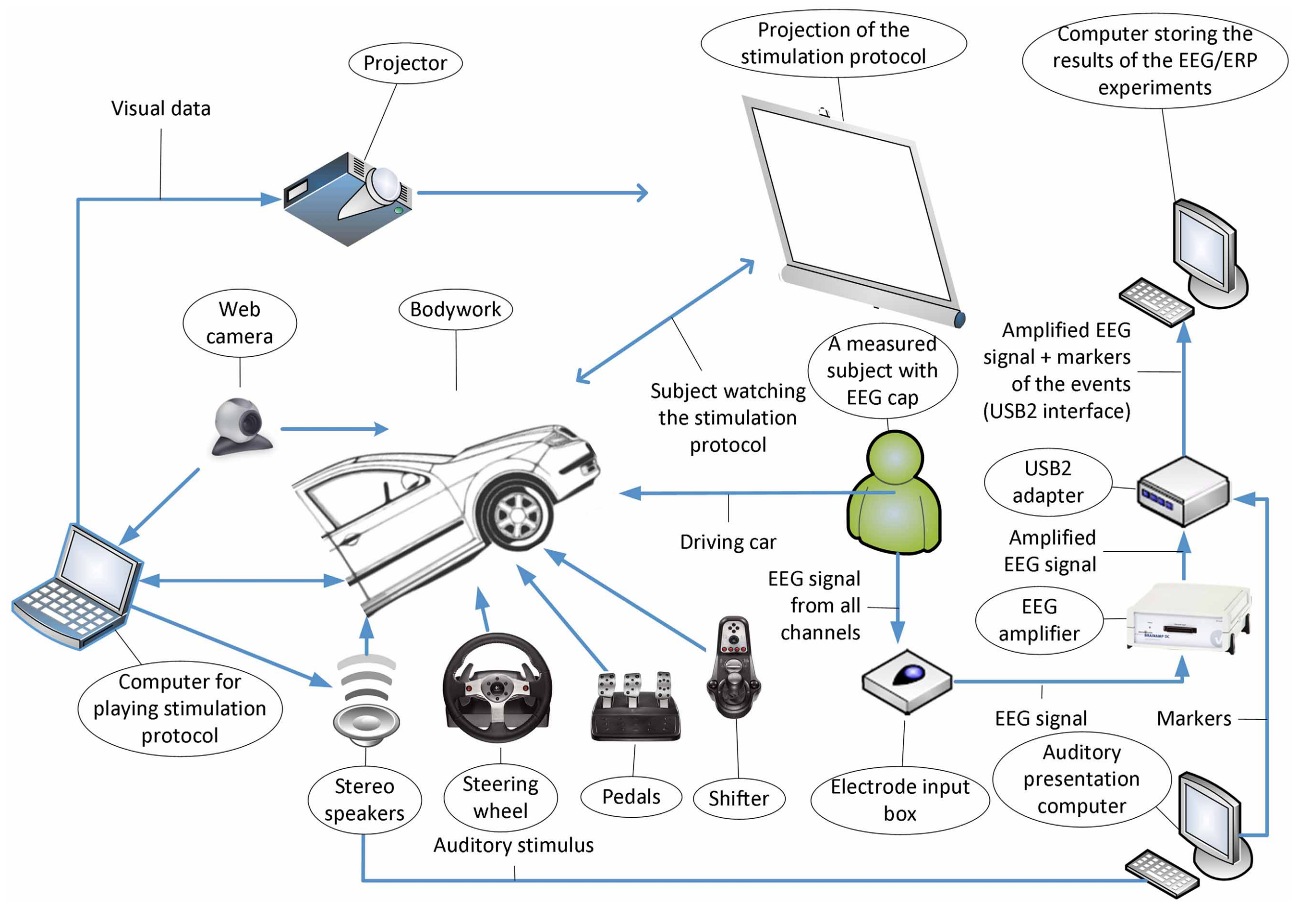

Projection of the stimulation protocol
Computer storing the results of the EEG/ERP experiments
Projector
Visual data
Web camera
Bodywork
A measured subject with EEG cap
Amplified EEG signal + markers of the events (USB2 interface)
Subject watching the stimulation protocol
USB2 adapter
Driving car
Amplified EEG signal
EEG signal from all channels
EEG amplifier
Computer for playing stimulation protocol
EEG signal
Markers
Auditory presentation computer
Stereo speakers
Steering wheel
Pedals
Shifter
Electrode input box
Auditory stimulus

FIGURE 4 | Drivers attention experimental setup (Bruha et al., 2013).

of the next method. The whole workflow process is divided into simple tasks—work steps. The work step includes one analytic method and requires following information (Mrvec, 2013):

- Name - identification of a work step
- Format - used data format as an input to a method
- Store - a decision, whether a result from a previous method should be stored
- Data - an input to a method e.g., data files or a name of previous workstep
- Method - a name of the used method with its parameters

A work unit representing a sequential workflow is composed of work steps. This solution allows creating more workflows with the same or different input data.

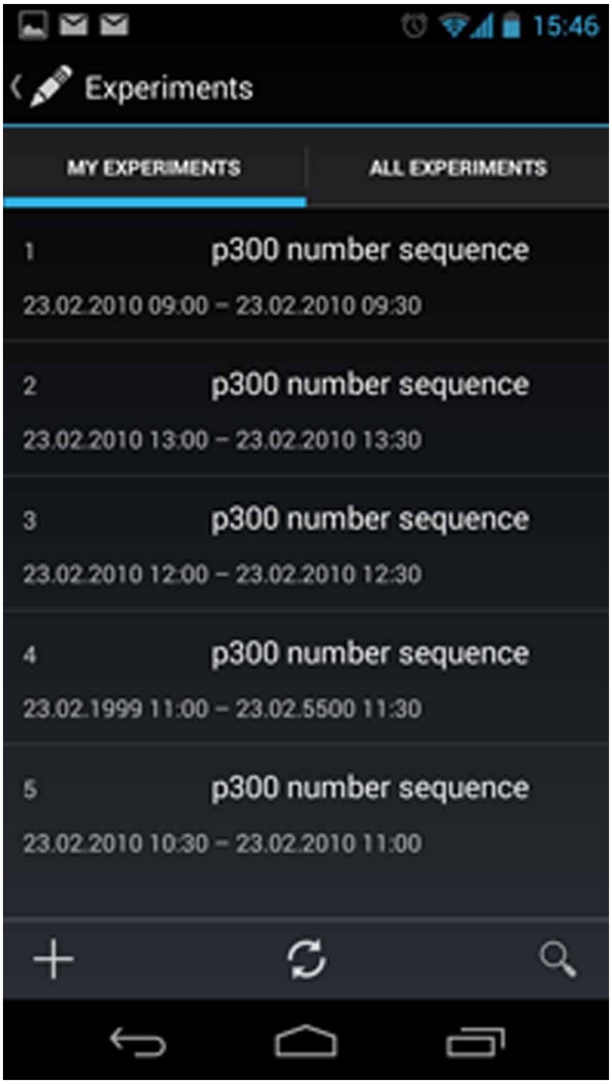
2.7. MOBILE AND OFFLINE PORTALS

The advantage of the EEG/ERP Portal is its accessibility from all computers connected to the Internet. Such solution is sufficient for collecting experiments performed in the laboratory. On the other hand, situations when a standard computer is not available are frequent. It includes situations when experiments are conducted outside the laboratory using a portable measuring device. In this case paper forms that are backward transferred to a central database are used. This process can be cumbersome, confusing, and error-prone. In addition, when data are collected electronically, they can be validated at the time of collection. It protects making logical errors or notational problems, and ensures that the required forms are complete.

Another use case is a situation when a researcher discusses experimental results with colleagues at workshops. He/she probably does not have desired data on hand. A mobile device used in everyday life, such as a mobile phone or tablet seems to be a practical solution for presenting experimental data.

With regards to the mentioned needs and difficulties a system for collecting experimental data/metadata running on mobile devices has been developed. The aim of this system is to serve as a mobile version of the EEG/ERP Portal. This mobile portal provides similar functionality as the common EEG/ERP Portal. The data from this device are synchronized with the data stored in the EEG/ERP Portal. This solution significantly reduces the usage of paper forms during experimenting. A preview of the mobile EEG/ERP Portal is shown in **Figure 5**.

An offline EEG/ERP Portal is a next useful system developed outside the EEG/ERP Portal. It is designed to be installed on computers or laptops without a permanent internet connection.



15:46

Experiments

MY EXPERIMENTS | ALL EXPERIMENTS

1 p300 number sequence
23.02.2010 09:00 – 23.02.2010 09:30

2 p300 number sequence
23.02.2010 13:00 – 23.02.2010 13:30

3 p300 number sequence
23.02.2010 12:00 – 23.02.2010 12:30

4 p300 number sequence
23.02.1999 11:00 – 23.02.5500 11:30

5 p300 number sequence
23.02.2010 10:30 – 23.02.2010 11:00

FIGURE 5 | The mobile system preview. The print screen shows a list of available scenarios. When a user clicks to a specific scenario, a detail piece of information appears. The top bar allows users to add a new scenario (using "+" button), search existing items (using magnifying glass), or refresh the list.

The offline EEG/ERP Portal became a part of the JERPA software tool (Jezek and Moucek, 2011a)—a desktop system for running signal processing methods and signal visualization. This system contains a powerful plug-in engine that enables installing signal processing methods as plug-ins. A server-client approach is used. A module that ensures an online access to experimental data stored in the EEG/ERP Portal implements a web service client. The EEG/ERP Portal represents the server side. Downloaded data are stored in an embedded database and they are available when the system gets offline. When a new experiment is added, it is synchronized with the EEG/ERP Portal when the system returns online. The stored data are ready to be processed by installed methods. An overview of the JERPA system is shown in **Figure 6**.

2.8. PROGRAMMABLE HARDWARE STIMULATOR

A programmable hardware stimulator was designed and developed for EEG/ERP experiments performed in our neuroinformatics laboratory. The main idea was to have a portable device which was very easy to use and did not require another hardware device (PC or laptop) needed for experiments. In addition, we also wanted to compare the results (e.g., timings and delays) from experiments in which a software stimulator had been used with the results from experiments in which a hardware stimulation device had been used.

To design and construct a first prototype we used a simple 8-bit microcontroller with an interrupt based firmware which works as a timed LED driver. The basic structure can be seen in the block diagram in **Figure 7**. This implementation was expanded step by step with different features like LED brightness, scalable distribution schemas, and new experiments predefinitions.

The finalized version provides a fully programmable setting of stimuli parameters in two scenarios. The first scenario is an implementation of the oddball protocol and the second scenario enables multi-source frequency stimulation. A simple GUI and serial port communication protocol were implemented.

At the moment we are working on a new version of the stimulator based on experience gained by using this prototype during experiments. It will provide a more comfortable user interface, a broader opportunity to set parameters for EEG/ERP experiments, and new possibilities in stimuli generation for auditory stimulation protocols.

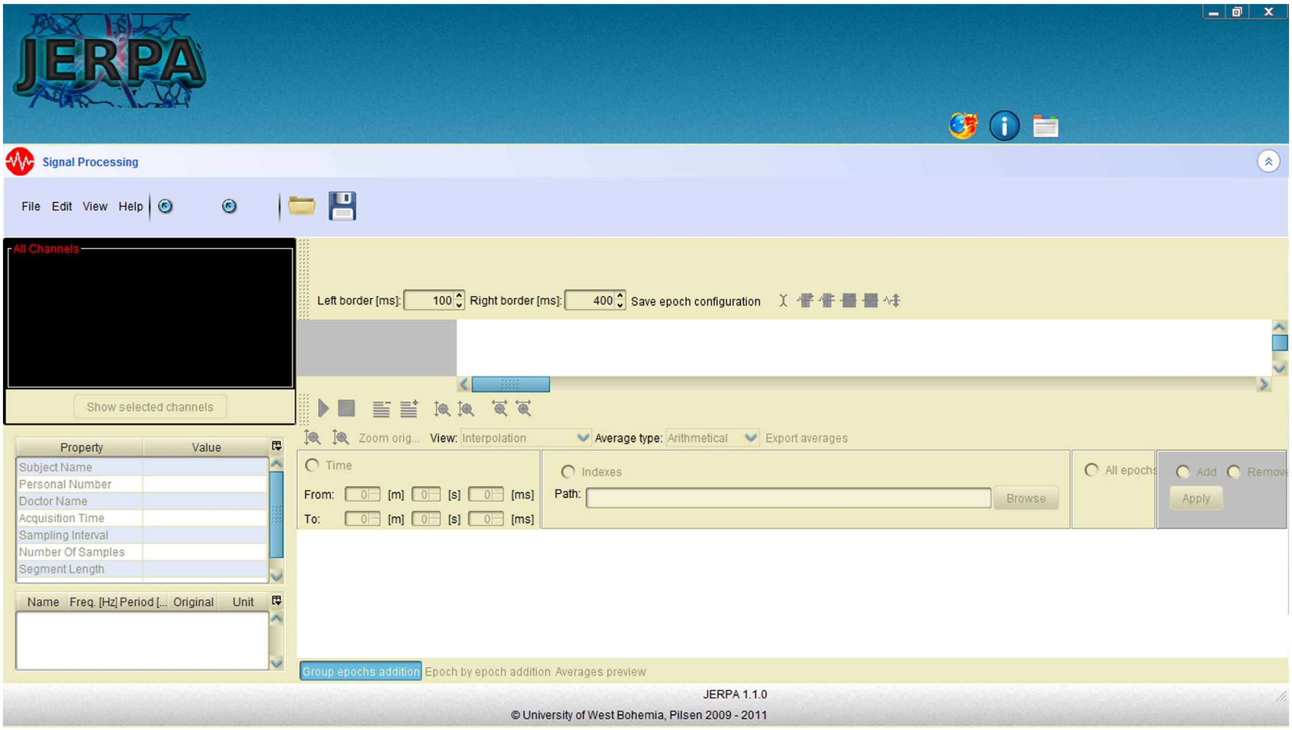
3. RESULTS

This section provides information about the current state of the proposed infrastructure and some implementation details of the parts of the infrastructure described in Section 2.

3.1. EEG/ERP PORTAL

The EEG/ERP portal is a central point of the complex architecture presented. It is a powerful tool intended to serve to a wide researcher's community. It facilitates management of experimental data; provide an interface for accessing them, and due to well-defined ontology it significantly helps in interpretation of experimental data. The portal interface is suitable not only for human readers who access it using a web browser, but due to an advanced web service endpoint it can be easily integrated with complementary tools. Such infrastructure is ready to be used not only in our laboratory but also by other interested researchers.

A core of the EEG/ERP Portal creates the Spring framework (Walls, 2011) that provides a comprehensive programing and configuration model for Java-based enterprise applications. The data layer of the EEG/ERP Portal uses the Oracle database system (Greenwald et al., 2007). The Hibernate framework (Bauer and King, 2006) ensures persistence of data transferred between the database and a Java-based application layer. The presentation layer is created by the Apache Wicket framework (Dashorst and Hillenius, 2008) that facilitates implementation by a system of reusable components written with plain Java and HTML. The privacy of stored data and integration with social networks as LinkedIn or Facebook are ensured by subcomponents of Spring: the Spring security framework and the Spring social framework.



Left border [ms]: 100 Right border [ms]: 400

View: Interpolation Average type: Arithmetical

From: 0 [m] 0 [s] 0 [ms]
To: 0 [m] 0 [s] 0 [ms]

Property	Value
Subject Name	
Personal Number	
Doctor Name	
Acquisition Time	
Sampling Interval	
Number Of Samples	
Segment Length	

Name	Freq. [Hz]	Period [...	Original	Unit

JERPA 1.1.0
© University of West Bohemia, Pilsen 2009 - 2011

FIGURE 6 | JERPA overview.

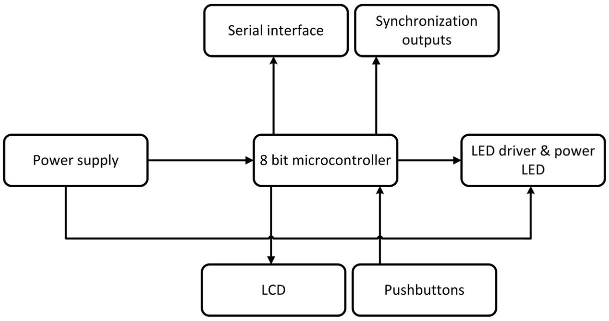


FIGURE 7 | Block diagram of the stimulator.

The Spring security framework uses a XML-based configuration for an authorized access to individual web pages. The Spring social framework provides a unified API to access various social networks.

The main integrated tool, the Semantic Framework (described in Section 2.3), is being developed as a single library. From the user's perspective, it is used as a black box with the input in the form of a collection of Java persistent objects and the output in the form of an ontology document. The ontology document can be serialized into several supported syntaxes [currently RDF/XML (W3C Consortium, 2004), OWL/XML (Motik and Patel-Schneider, 2008), Turtle (W3C Consortium, 2008), and abbreviated OWL/XML formats are supported]. The Semantic Framework is controlled by a build-in timer. The timer calls the Semantic Framework API in regular intervals. The API generates the ontology document from the stored experiments and saves this document to a temporary file. When any document request appears, the temporary file containing the current set of stored experiments is immediately available.

External tools work independently of the EEG/ERP Portal. The EEG/ERP Portal provides an interface for accessing stored experiments using Web Services technology; RESTfull (Richardson and Ruby, 2007) and SOAP (Snell et al., 2002) web services are used. These web services are secured by user credentials and provide several methods to access user's experiments including raw data, metadata and experimental scenarios. An interested client only implements a web service client. Several tools presented in this paper such as the offline EEG/ERP Portal or mobile EEG/ERP Portal implement the web service client to access experimental data and so prove the validity of the approach presented.

3.2. DATA MODELS AND ONTOLOGIES

The EEG/ERP Portal was registered as a neuroscience resource within the NIF at the level 2.5; this level allows users direct access to the services implemented within the EEG/ERP Portal. Privacy and security of the stored data are guaranteed by data and metadata anonymization. Currently, the data about tested subjects, raw experimental data, data related to used hardware, experimental protocol (scenario), and other experimental parameters (e.g., length of recording) are accessible via NIF (Bruha and Moucek, 2012). The Portal is intended to be registered at the 3rd level of NIF registration schema. This step will be accompanied by the annotation of the Portal data and metadata using the OEN ontology.

Figure 8 shows how to describe the experimental set-up using terms (labels/names in bold) from OBI (obi_xxxxxxx), NEMO (NEMO_xxxxxx), OEN (oen_xxxxxxx) and relations (dashed arrow, label, and id) (Bruha et al., 2013).

3.3. SIGNAL PROCESSING METHODS

From researchers' point of view, our signal processing methods infrastructure consists of the following commercial elements: MATLAB (including EEGLAB Delorme and Makeig, 2004 plug-in) MATLAB (2012), the BrainVision Recorder and the BrainVision Analyzer applications (BrainProducts, 2012). Furthermore, the following elements of our infrastructure are freely available: Lastwave (Bacry, 2012), JERPA (Jezek and Moucek, 2011a), the EEG/ERP Portal, the EEGDSP java library (described below), and the EEG Data Processor (Jezek and Moucek, 2013b). **Figure 9** shows this infrastructure including relationships between the elements.

Researchers can access and work with all elements except the EEG amplifier and the EEGDSP.jar library. The reason is that both of them need an interface that provides them functionality. In case of the EEG amplifier, the interface is the Brain Vision Recorder. In case of the EEGDSP.jar library, the interface is the EEG Data Processor or all programing languages which can call external Java libraries.

The EEGDSP library was implemented in the Java language and includes basic methods and approaches for discrete signal processing: wavelet transform, matching pursuit algorithm, fast ICA, FIR filters (low pass, high pass, band pass, band reject), window functions (BarlettHann, Barlett, BlackmanHarris, BlackmanNuttalll, Blackman, Bohman, Cosine, FlatTop, Gauss, Hamming, Hanning, Kaiser, Lonczos, Nuttall, Parzen, Rectangular, Triangular, and Tukey), and Hilbert-Huang transform. The implementation of the Hilbert-Huang transform uses the modified HHT algorithm (Section 2.5) to detect ERP components in the EEG signal (there was no free or commercial library with implemented modified HHT before). To facilitate the usage of the discrete signal processing methods by researchers and services, the application Electroencephalography Data Processor (Section 2.3) was implemented. The data processing feature is powered by the EEGDSP library.

3.4. WORFLOWS

The workflow management system is currently under development. Any workflow is described by an XML file. We chose the XML format since it is independent of the used platform and programming language. An example of the workflow description is given below (Mrvec, 2013).

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow name="Workflow">
 <workunit name="Experiment1">
   <workstep name="SimpleFile" format="KIV_FORMAT"
                store="false">
     <data>data1.eeg</data>
     <data>data1.vhdr</data>
     <method params="01,100,Cz,FAST_DAUBECHIES_2">
                 DWTPlugin-1.0.0</method>
   </workstep>
   <workstep name="SimpleDouble" format="DOUBLE_FORMAT"
                store="true">
     <data>Experiment1_SimpleFile</data>
     <method params="01,1000,Cz,COMPLEX_GAUSSIAN,
                  1,1,1,14000,14000">
     CWTPlugin-1.0.0
     </method>
   </workstep>
 </workunit>
 <workunit name="Experiment2">
   <workstep name="SimpleFile" format="KIV_FORMAT"
                store="true">
     <data>data2.eeg</data>
     <data>data2.vhdr</data>
     <method params="01,100,Cz,FAST_DAUBECHIES_2">
                 DWTPlugin-1.0.0</method>
   </workstep>
 </workunit>
</workflow>
```

This XML file is generated while the user creates a workflow (he/she selects methods, defines values of their parameters, and puts them into analytic pipelines). When the user finishes his/her workflow, the XML file is transferred to a processing unit that is responsible for parsing the file and calling required methods. This approach allows changing a source of analytic methods (e.g., EEG Data Processor, Matlab scripts, or local libraries) without changing a generation process of the descriptive file. It is only necessary to change the processing unit and a graphic user interface.

Since the used analytic methods have various input/output parameter types, it is necessary to ensure their compatibility in sequential workflows. It means that the output from a previous method and the input to a next method must match. We ensured the syntactic compatibility by comparison of input/output parameters types. Each used method has a definition of input/output parameters by the XML file attached to the methods.

However, for well-designed workflows, ensuring syntactic compatibility is a necessary, but a single step. The used methods have to be also connected correctly in terms of their semantics (if their connection makes sense or not). Therefore, we will focus on designing the semantic compatibility in the future.

3.5. MOBILE AND OFFLINE PORTALS

Because of difficulties with unavailability of standard computers in many environments we developed a mobile version of the EEG/ERP Portal that is able to fully substitute the EEG/ERP Portal when experiments are performed outside the laboratory. This solution profits from rising popularity of mobile devices such as tablets or mobile phones. The presented implementation can be extended to enable work with other electrophysiological databases. It will result in a domain independent system using a customizable user layout.

When the user works on a portable device as a laptop in environments when the internet connection is not available (as hospitals or other institutions outside the laboratory), the offline version of the EEG/ERP Portal is available.

From the implementation point of view, the mobile EEG/ERP Portal contains a set of forms where the user can fill in metadata describing an experiment. The set of metadata is equivalent to

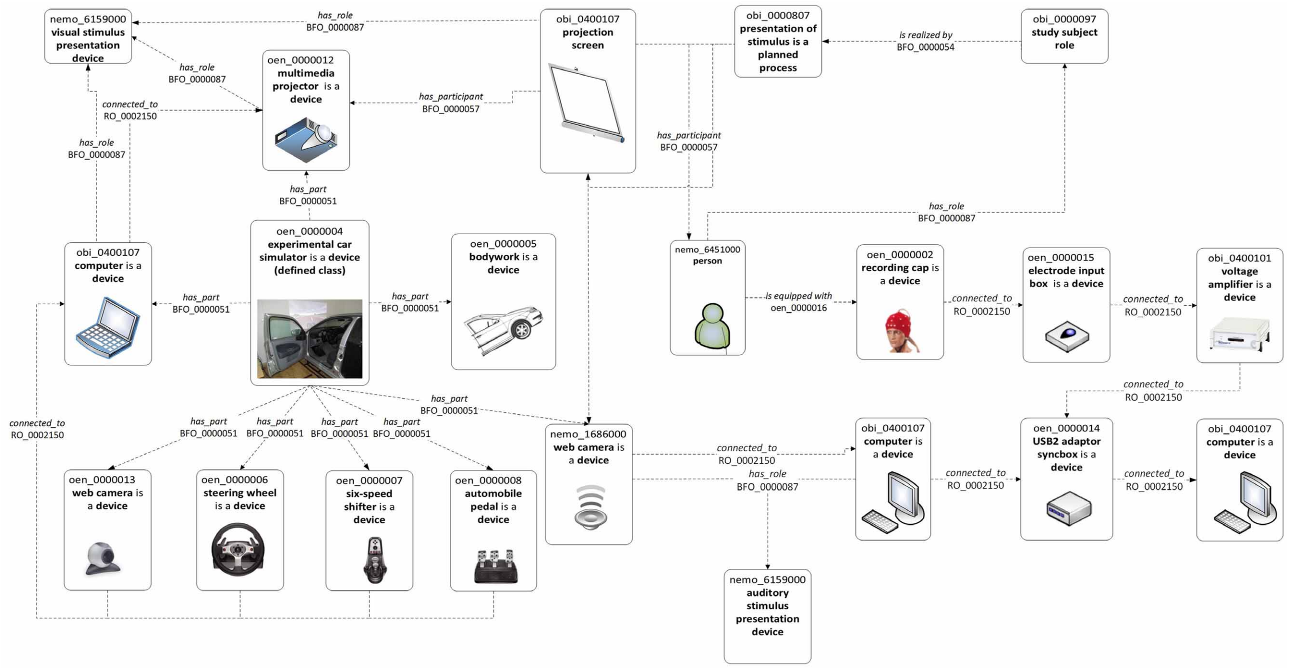


FIGURE 8 | EEG/ERP Portal device knowledge model (Bruha et al., 2013).

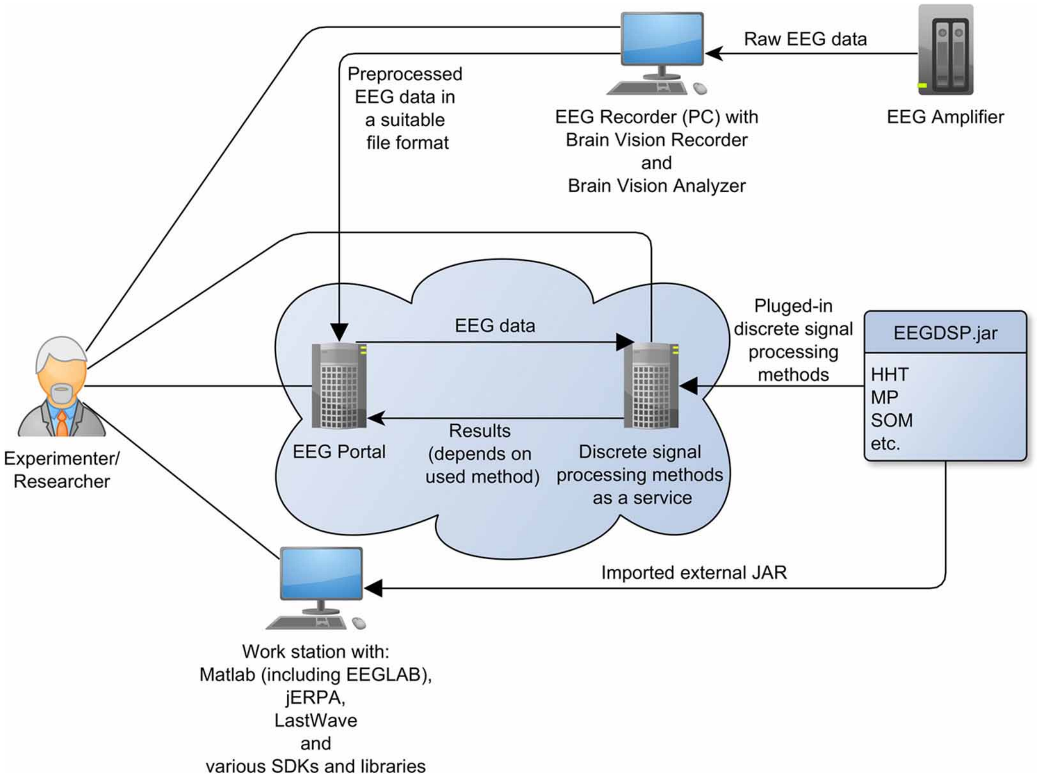

Raw EEG data
Preprocessed EEG data in a suitable file format
EEG Recorder (PC) with Brain Vision Recorder and Brain Vision Analyzer
EEG Amplifier
EEG data
Pluged-in discrete signal processing methods
EEGDSP.jar
HHT
MP
SOM
etc.
Experimenter/ Researcher
EEG Portal
Results (depends on used method)
Discrete signal processing methods as a service
Imported external JAR
Work station with: Matlab (including EEGLAB), jERPA, LastWave and various SDKs and libraries

FIGURE 9 | Signal processing within the infrastructure.

the set of metadata that the user can fill in the common EEG/ERP Portal. The communication of both the mobile EEG/ERP Portal and web EEG/ERP Portal is ensured using RESTfull web services. Server-client architecture is used. The server part is implemented in the EEG/ERP Portal. The server provides access to the database and sends data to the client implemented inside the mobile device. The communication between the server and client is secured using the SSL protocol. User credentials are required; the EEG/ERP Portal user account is used to verify the client (Jezek and Moucek, 2013a).

3.6. PROGRAMMABLE HARDWARE STIMULATOR

The hardware stimulator described in Section 2.8 was designed and tested for elicitation of the visual P3 component. The arrangement of a typical experiment and connection of the developed stimulator to the ERP recording system are presented in **Figure 10**. The standard oddball task was used to verify the functionality of the proposed hardware stimulator. In this task, red and green LEDs representing non-target and target stimuli were randomly switched on and off for a period of 0.5 s. The probability of the target stimuli (i.e., the green LED was switched on) was set up to 0.2. Currently, the hardware stimulator was successfully used for stimulation of 15 tested subjects.

3.7. SOFTWARE TOOLS LICENCE INFORMATION

The tools developed at our department, including the EEG/ERP Portal, the mobile EEG/ERP Portal and the Semantic Framework, are distributed under the Apache License 2.0. The EEGDSP library, JERPA, and the EEG Data processor are distributed under the GNU General Public License v.3. All the tools mentioned above are hosted in GitHub repositories. The EEG/ERP Portal is available under the INCF group at https://github.com/INCF/eeg-database. The EEG/ERP Portal is running on http://eegdatabase.kiv.zcu.cz. The following libraries, including the EEGDSP library, JERPA, the mobile EEG/ERP Portal and the Semantic Framework are hosted under the neuroinformatics group and available at https://github.com/NEUROINFORMATICS-GROUP-FAV-KIV-ZCU.

4. DISCUSSION

There are a lot of difficulties with collection, storage, management and interpretation of electrophysiological experimental data and metadata. Any complex software and hardware infrastructure supporting research and researchers in this field can contribute to efficiency and effectiveness of researchers' work.

This paper shortly introduced the infrastructure for research in electrophysiology that has been continuously built in the neuroinformatics laboratory at the Department of Computer Science and Engineering, University of West Bohemia. Over time the parts of the infrastructure have become more general with the potential to serve to the wider scientific community. Of course, there are still many infrastructural parts that need to be changed, finished or even only properly designed.

The central point of the described infrastructure, the EEG/ERP Portal, serves as a data management tool that provides services for other supplementary tools. Because the relational database is currently used as persistence storage, we are facing difficulties with storing heterogeneous experimental data. Our next step leads to the usage of a NoSQL database instead of the relational one. Currently we test Elasticsearch for its full text search capabilities. The future direction is to provide a domain independent metadata structure that enables to store various experimental data from laboratories.

Because some experiments are conducted outside the laboratory, the mobile version of the EEG/ERP Portal was presented. In response to positive feedbacks the next significant step is to provide an extension of this system independent of the EEG/ERP Portal. Such extended system will communicate with other domain independent electrophysiological databases. The layout of this system will be generated automatically as proposed in Jezek et al. (2013a). odML as a unified metadata format will ensure a server-client data transfer. Using a NoSQL database also means to modify the supplementary tools, the offline EEG/ERP Portal, and the JERPA system.

The ontology development was first focused on the experimental data and metadata stored in the EEG/ERP Portal. Currently, the emerging OEN ontology is not a specific ontology describing just the data and metadata stored in the EEG/ERP Portal; its concept is open for any neurophysiological needs. Nevertheless, the EEG/ERP Portal will be the first use case fully described by this ontology. This will also help to fulfil NIF requirements for the registration at the 3rd level of the NIF portal. Then the data and metadata from the EEG/ERP Portal will be fully accessible to other communities and research groups via the NIF interface.

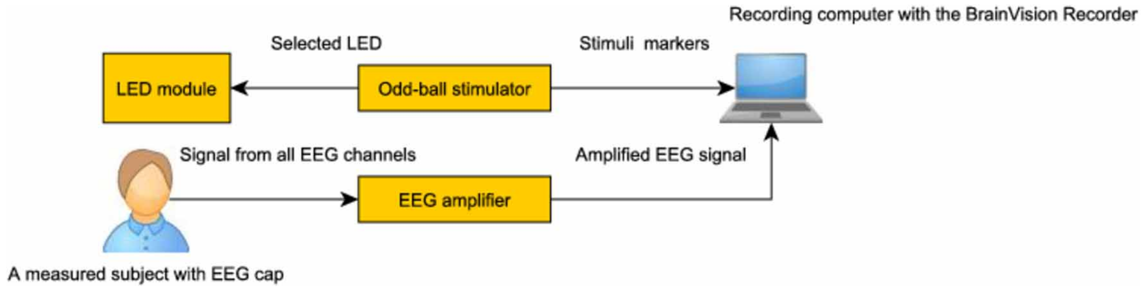

Recording computer with the BrainVision Recorder
Selected LED
Stimuli markers
LED module
Odd-ball stimulator
Signal from all EEG channels
Amplified EEG signal
EEG amplifier
A measured subject with EEG cap

FIGURE 10 | Experimental usage of the programmable hardware stimulator.

Various tools for EEG signal processing have been used by our research group. Certainly, researchers would benefit from a possibility of having all methods at the same place. The EEG Data Processor seems to be an appropriate solution. However, there are some issues to be solved. We are not able to implement all methods that researchers need. Therefore, we plan, according to the best principles in software engineering, to implement a common interface which encloses the implemented methods and gives researchers a unified way to use them on a source code level; then it is easy to include custom methods into the EEGDSP implementation. In the medium term, we will focus on distributed computing as well as on load balancing.

For a complex analysis, more methods are combined sequentially or in parallel; the workflows are created. The presented prototype of the workflow management system is still under development. The proposed description of workflows and the used XML language bring the independence of the methods used and the programming language in which they are written. In the future, the workflow system will be integrated into the EEG/ERP Portal.

The programmable hardware stimulator is applicable to a variety of experiments. The modular design of the firmware allows users to modify stimulation protocols according to experimenters' requirements easily. In the near future, a miniaturized stimulator using 32bit MCU will be developed. Moreover, this stimulator will allow users to apply a larger set of stimulation methods and thus it can be used in a larger number of experimental protocols (e.g., our research group plans to use it for BCI experiments and for the stimulation of the mouse brain).

We are aware that the current state of the infrastructure is intended for storing, maintenance and analysis of EEG/ERP waveforms and related metadata. On the other hand, long-term work on this initial infrastructure has helped the research group to understand heterogeneity of neurophysiological data, limitations of the proposed methodology and also limitations of used technologies. The future work of the research group thus includes changes in methodological concepts (e.g., usage of international standards for data formats and ontologies or definition of a wider collection of use cases in neurophysiology) and technologies (e.g., technological solutions supporting more flexible organization of data). On the other hand, there is also the danger that the proposed infrastructure could be too general. It results in difficult implementation, configuration, and too complex and demanding user interface. Aware of both the difficulties and dangers, too high specialization and/or too high abstraction of the system, the research group intends to continuously improve the existing infrastructure for specific purposes. In parallel, it plans to extend the ability of the existing infrastructure to store and process a larger variety of neurophysiological data by following the international standardization efforts and by respecting the needs of researchers. Some of these methodological and technological steps (OEN ontology, odML format, NoSQL database) are already described in this article.

4.1. DATA SHARING

Our catalog server connected to INCF Dataspace and a node for eeg/erp domain (a subnode of the catalog server) were established. Specifically, the node named "cz.zcu.eeg" contains the collection named "experiments" with a set of subcollections grouped according to experimental scenarios. These subcollections contain experimental data divided into individual sessions. Metadata are stored in CSV files. The node server is synchronized with the data in the EEG/ERP Portal in regular intervals using an implemented timer. It ensures availability of up-to-date experimental data. The data stored in the EEG/ERP Portal are also shared via NIF.

ACKNOWLEDGMENTS

Funding: The work was supported by the UWB grant SGS-2013-039 Methods and Applications of Bio- and Medical Informatics and by the European Regional Development Fund (ERDF), Project "NTIS - New Technologies for Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090.

REFERENCES

Aviyente, S., Bernat, E. M., Malone, S. M., and Iacono, W. G. (2006). Analysis of event related potentials using PCA and matching pursuit on the time-frequency plane. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 1, 2454–2457. doi: 10.1109/IEMBS.2006.259590

Bacry, E. (2012). *Lastwave*. Available online at: http://www.cmap.polytechnique.fr/~bacry/LastWave/

Bauer, C., and King, G. (2006). *Java Persistence with Hibernate*. Revised Edn. Shelter Island, NY: Manning.

Benar, C. G., Papadopoulo, T., and Clerc, M. (2007). Topography-time-frequency atomic decomposition for event-related M/EEG signals. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2007, 5461–5464. doi: 10.1109/IEMBS.2007.4353581

Borlawsky, T. B., Lele, O., Jensen, D., Hood, N. E., and Wewers, M. E. (2011). Enabling distributed electronic research data collection for a rural appalachian tobacco cessation study. *J. Am. Med. Inform. Assoc.* 18(Suppl. 1), i140–i143. doi: 10.1136/amiajnl-2011-000354

BrainProducts. (2012). *BrainVision Analyzer 2*. Available online at: http://www.brainproducts.com/productdetails.php?id=17

Brinkman, R., Courtot, M., Derom, D., Fostel, J., He, Y., Lord, P., et al. (2010). Modeling biomedical experimental processes with OBI. *J. Biomed. Sem.* 1(Suppl. 1), S7+. doi: 10.1186/2041-1480-1-S1-S7

Bruha, P., and Moucek, R. (2012). "Portal for research in electrophysiology - data integration with neuroscience information framework," in *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on* (Chongqing), 1099–1103. doi: 10.1109/BMEI.2012.6513049

Bruha, P., Papez, V., Bandrowski, A., Grewe, J., Moucek, R., Tripathy, S., et al. (2013). The ontology for experimental neurophysiology: a first step toward semantic annotations of neurophysiology data and metadata. *Front. Neuroinform. Conference Abstract: Neuroinformatics 2013*, 26. doi: 10.3389/conf.fninf.2013.09.00026

Carmen. (2013). *Carmen Portal*. Available online at: http://www.carmen.org.uk/

Carpenter, G. A., and Grossberg, S. (1987). Art 2: self-organization of stable category recognition codes for analog input patterns. *Appl. Opt.* 26, 4919–4930. doi: 10.1364/AO.26.004919

Carreiras, C., Silva, H., Lourenço, A., and Fred, A. L. N. (2013). "Storagebit - a metadata-aware, extensible, semantic and hierarchical database for biosignals," in *HEALTHINF*, eds D. Stacey, J. Sol Casals, A. L. N. Fred, and H. Gamboa (Barcelona: SciTePress), 65–74.

Ciniburk, J. (2011). *Hilbert-Huang Transform for ERP Detection*. Ph.D. thesis, Faculty of Applied Sciences, University of West Bohemia, Univerzitni 22, 306 14 Pilsen.

Cong, F., Sipola, T., Huttunen-Scott, T., Xu, X., Ristaniemi, T., and Lyytinen, H. (2009). Hilbert-huang versus morlet wavelet transformation on mismatch negativity of children in uninterrupted sound paradigm. *Nonlinear Biomed. Phys.* 3:1. doi: 10.1186/1753-4631-3-1

Dashorst, M., and Hillenius, E. (2008). *Wicket in Action*. Greenwich, CT: Manning Publications Co.

Delorme, A., and Makeig, S. (2004). Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *J. Neurosci. Methods* 134, 9–21. doi: 10.1016/j.jneumeth.2003.10.009

Dien, J. (2012). Applying principal components analysis to event-related potentials: a tutorial. *Dev. Neuropsychol.* 37, 497–517. doi: 10.1080/87565641.2012.697503

Dou, D., Frishkoff, G., Rong, J., Frank, R., Malony, A., and Tucker, D. (2007). "Development of neuroelectromagnetic ontologies(nemo): a framework for mining brainwave ontologies," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '07*, (New York, NY), 270–279. doi: 10.1145/1281192.1281224

EEG/ERP Portal. (2013). *EEG ERP Portal.* Available online at: http://eegdatabase.kiv.zcu.cz/

Ferrando, S. E., Kolasa, L. A., and Kovacevic, N. (2002). Algorithm 820: a flexible implementation of matching pursuit for gabor functions on the interval. *ACM Trans. Math. Softw.* 28, 337–353. doi: 10.1145/569147.569151

Gardner, D., Akil, H., Ascoli, G. A., Bowden, D. M., Bug, W. J., Donohue, D. E., et al. (2008). The neuroscience information framework: a data and knowledge environment for neuroscience. *Neuroinformatics* 6, 149–160. doi: 10.1007/s12021-008-9024-z

G-Node. (2013). *G-Node Tools and Services.* Available online at: http://www.g-node.org/

Greenwald, R., Stackowiak, R., and Stern, J. (2007). *Oracle Essentials - Oracle Database 11g: What You Need to Know About Oracle Database Architecture and Features: Covers Oracle Database 11g and Earlier Releases.* 4th Edn. Sebastopol: O'Reilly.

Grewe, J., Wachtler, T., and Benda, J. (2011). A bottom-up approach to data annotation in neurophysiology. *Front. Neuroinform.* 5:16. doi: 10.3389/fninf.2011.00016

Harris, P. A. (2012). Research electronic data capture (redcap) - planning, collecting and managing data for clinical and translational research. *BMC Bioinform.* 13:A15. doi: 10.1186/1471-2105-13-S12-A15

HDF Group. (2013). *Hierarchical Data Format*. Available online at: http://www.hdfgroup.org/

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., et al. (1998). The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. A Math. Phys. Eng. Sci.* 454, 903–995. doi: 10.1098/rspa.1998.0193

INCF. (2013). *International Neuroinformatics Coordinating Facility*. Available online at: http://www.incf.org/

INCF Group. (2013). *INCF Dataspace*. Available online at: http://www.incf.org/resources/data-space/

Jezek, P., and Moucek, R. (2011a). "Integration of signal processing methods into eeg/erp system," in *HEALTHINF 2011 - Proceedings of the International Conference on Health Informatics*, eds V. Traver, A. L. N. Fred, J. Filipe, and H. Gamboa (Rome: SciTePress), 563–566.

Jezek, P., and Moucek, R. (2011b). "Semantic web in eeg/erp portal: ontology development and nif registration," in *Biomedical Engineering and Informatics (BMEI), 2011 4th International Conference on* (Shanghai), Vol. 4 2058–2062.

Jezek, P., and Moucek, R. (2012a). "Semantic web in eeg/erp portal - extending of data layer using java annotations," in *HEALTHINF 2012 - Proceedings of the International Conference on Health Informatics*, eds E. Conchon, C. M. B. A. Correia, A. L. N. Fred, and H. Gamboa (Vilamoura, Algarve: SciTePress), 350–353.

Jezek, P., and Moucek, R. (2012b). System for EEG/ERP Data and metadata storage and management. *Neural Netw. World* 22, 277–290.

Jezek, P., and Moucek, R. (2013a). "EEG/ERP portal for android platform," in *Front. Neuroinform. Conference Abstract: Neuroinformatics 2013* (Stockholm), 46. doi: 10.3389/conf.fninf.2013.09.00046

Jezek, P., and Moucek, R. (2013b). "Electroencephalography data processor - framework for running signal processing methods," in *HEALTHINF*, eds D. Stacey, J. Solé-Casals, A. L. N. Fred, and H. Gamboa (Barcelona: SciTePress), 357–361.

Jezek, P., Moucek, R., and Danek, J. (2014). "Mongodb for electrophysiology experiments," in *HEALTHINF*, (SciTePress).

Jezek, P., Moucek, R., Le Franc, Y., Wachtler, T., and Grewe, J. (2013a). "Framework for automatic generation of graphical layout compatible with multiple platforms," in *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*, eds C. Kelleher, M. M. Burnett, and S. Sauer (San-Jose: IEEE), 193–194.

Jezek, P., Stebeták, J., Bruha, P., and Moucek, R. (2013b). "Model of software and hardware infrastructure for electrophysiology," in *HEALTHINF*, eds D. Stacey, J. Solé-Casals, A. L. N. Fred, and H. Gamboa (Barcelona: SciTePress), 352–356.

Lotte, F., Congedo, M., Lcuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for eeg-based braincomputer interfaces. *J. Neural Eng.* 4:R1. doi: 10.1088/1741-2560/4/2/R01

Luck, S. (2005). "An introduction to the event-related potential technique," in *Cognitive Neuroscience* (Cambridge, MA: MIT Press).

Makeig, S., Jung, T.-P., Bell, A. J., Ghahremani, D., and Sejnowski, T. J. (1997). Blind separation of auditory event-related brain responses into independent components. *Proc. Natl. Acad. Sci. U.S.A.* 94, 10979–10984. doi: 10.1073/pnas.94.20.10979

Mallat, S., and Zhang, Z. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Trans. Signal Process.* 41, 3397–3415. doi: 10.1109/78.258082

MATLAB. (2012). *Version 7.14.0.739 (R2012a)*. Natick, MA: The MathWorks Inc.

Motik, B., and Patel-Schneider, P. (2008). *Owl 2 Web Ontology Language: Xml Serialization.* Cambridge, MA: World Wide Web Consortium.

Mrvec, R. (2013). *System Workflows V EEG/ERP Portalu (Workflows in EEGbase)*. Technical report, Department of Computer Science and Engineering, University of West Bohemia, Pilsen, Czech Republic.

NIF. (2013). *Neuroscience Information Framework*. Available online at: http://www.neuinfo.org/

Picton, T. W., Lins, O. G., and Scherg, M. (1995). "The recording and analysis of event-related potentials," in *Handbook of Neuropsychology, Vol. 10*, eds F. Boller, and J. Grafman (Amsterdam: Elsevier), 3–73.

Prokop, T. (2013). *Methods of Evaluation of Electrophysiological Experiments*. Ph.D. thesis, University of West Bohemia, Faculty of Applied Sciences, Univerzitni 22, 306 14 Pilsen.

Quiroga, R., and Garcia, H. (2003). Single-trial event-related potentials with wavelet denoising. *Clin. Neurophys.* 114, 376–390. doi: 10.1016/S1388-2457(02)00365-6

Richardson, L., and Ruby, S. (2007). *RESTful Web Services*. Beijing: O'Reilly.

Robbins, K. A. (2012). Eegvis: a matlab toolbox for browsing, exploring, and viewing large datasets. *Front. Neuroinform.* 6:17. doi: 10.3389/fninf.2012.00017

Rondik, T. (2010). *Methods of ERP Signals Processing*. Technical report, Department of Computer Science and Engineering, University of West Bohemia, Pilsen, Czech Republic.

Snell, J., Tidwell, D., and Kulchenko, P. (2002). *Programming Web Services with SOAP*. Sebastopol, CA: O'Reilly and Associates, Inc.

Stacey, D., Solé-Casals, J., Fred, A. L. N., and Gamboa, H., (eds.). (2013). *HEALTHINF 2013 in Proceedings of the International Conference on Health Informatics* (Barcelona, Spain: SciTePress).

Teeters, J., Harris, K., Millman, K., Olshausen, B., and Sommer, F. (2008). Data sharing for computational neuroscience. *Neuroinformatics* 6, 47–55. doi: 10.1007/s12021-008-9009-y

Turner, J. A., Lane, S. R., Bockholt, H. J., and Calhoun, V. D. (2011). The clinical assessment and remote administration tablet. *Front. Neuroinform.* 5:31. doi: 10.3389/fninf.2011.00031

Vareka, L. (2012). "Matching pursuit for p300-based brain-computer interfaces," in *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on* (Prague), 513–516. doi: 10.1109/TSP.2012.6256347

Vareka, L., and Mautner, P. (2012). "The event-related potential data processing using art 2 network," in *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on* (Chongqing), 605–609. doi: 10.1109/BMEI.2012.6513044

Vareka, L., and Mautner, P. (2013). "Off-line analysis of the p300 event-related potential using discrete wavelet transform," in *Telecommunications and Signal Processing (TSP), 2013 36th International Conference on* (Rome), 569–572. doi: 10.1109/TSP.2013.6613998

W3C Consortium. (2004). "RDF/XML Syntax Specification (Revised)," in *Technical Report, W3C* (Cambridge, MA).

W3C Consortium. (2008). Turtle - terse RDF triple language, W3C team submission. Available online at: http://www.w3.org/TeamSubmission/turtle/

Walls, C. (2011). *Spring in Action*. Shelter Island, NY: Manning.

Xiang, Z., Courtot, M., Brinkman, R., Ruttenberg, A., and He, Y. (2010). OntoFox: web-based support for ontology reuse. *BMC Res. Notes* 3:175+. doi: 10.1186/1756-0500-3-175

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 30 December 2013; accepted: 19 February 2014; published online: 07 March 2014.

Citation: Mouček R, Ježek P, Vařeka L, Řondík T, Brůha P, Papež V, Mautner P, Novotný J, Prokop T and Štěbeták J (2014) Software and hardware infrastructure for research in electrophysiology. Front. Neuroinform. ***8****:20. doi: 10.3389/fninf.2014.00020*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 29 April 2014
doi: 10.3389/fninf.2014.00040

NeuroElectro: a window to the world's neuron electrophysiology data

Shreejoy J. Tripathy[1,2*†], **Judith Savitskaya**[1†], **Shawn D. Burton**[1,2], **Nathaniel N. Urban**[1,2] and **Richard C. Gerkin**[1,2*†]

[1] Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, PA, USA
[2] Center for the Neural Basis of Cognition, Carnegie Mellon University, Pittsburgh, PA, USA

Edited by:
John Van Horn, University of California at Los Angeles, USA

Reviewed by:
Richard Coppola, National Institute of Mental Health, USA
Andrei Irimia, University of Southern California, USA

***Correspondence:**
Shreejoy J. Tripathy, 177 Michael Smith Laboratories, University of British Columbia, 2185 East Mall, BC, V6T 1Z4, Canada
e-mail: stripat3@gmail.com;
Richard C. Gerkin, School of Life Sciences, Arizona State University, PO Box 874501, 550 E. Orange St., Tempe, AZ 85281, USA
e-mail: rgerkin@asu.edu

†Present address:
Shreejoy J. Tripathy, Centre for High-Throughput Biology and Department of Psychiatry, University of British Columbia, BC, Canada;
Judith Savitskaya, Graduate Program in Bioengineering, University of California, Berkeley and University of California, San Francisco, CA, USA;
Richard C. Gerkin, School of Life Sciences, Arizona State University, Tempe, AZ, USA

The behavior of neural circuits is determined largely by the electrophysiological properties of the neurons they contain. Understanding the relationships of these properties requires the ability to first identify and catalog each property. However, information about such properties is largely locked away in decades of closed-access journal articles with heterogeneous conventions for reporting results, making it difficult to utilize the underlying data. We solve this problem through the NeuroElectro project: a Python library, RESTful API, and web application (at http://neuroelectro.org) for the extraction, visualization, and summarization of published data on neurons' electrophysiological properties. Information is organized both by neuron type (using neuron definitions provided by NeuroLex) and by electrophysiological property (using a newly developed ontology). We describe the techniques and challenges associated with the automated extraction of tabular electrophysiological data and methodological metadata from journal articles. We further discuss strategies for how to best combine, normalize and organize data across these heterogeneous sources. NeuroElectro is a valuable resource for experimental physiologists attempting to supplement their own data, for computational modelers looking to constrain their model parameters, and for theoreticians searching for undiscovered relationships among neurons and their properties.

Keywords: neuroinformatics, electrophysiology, database, text-mining, metadata, API, machine learning, natural language processing

1. INTRODUCTION

Brains achieve efficient function through implementing a division of labor, in which different types of neurons serve distinct functional and computational roles. One striking way in which neuron types differ is in their electrophysiology properties. Though the electrophysiology of many neuron types has been previously characterized and documented across decades of research, these data exist across thousands of journal articles, making cross-study neuron-to-neuron comparisons difficult.

Neurophysiology lacks a centralized resource where consensus data on basic physiological measurements from many neuron types and studies are accessible for reference and subsequent meta-analyses. For example, though it is common for neurophysiologists to measure and report neuronal measurements such as resting membrane potential and input resistance, there is not a public database which compiles this information. In other domains of neuroscience such efforts have made more progress. In the domain of neuroanatomical connectivity, information on connectivity between different brain regions is being compiled by experts at the Brain Architecture Management System project (BAMS) across thousands of publications (Bota et al., 2005). Parallel to this effort is the WhiteText Project, which addresses a complementary goal by algorithmically mining brain region connectivity statements from journal abstracts using biomedical natural language processing (bioNLP) methods (French et al., 2009, 2012). Similarly, in the domain of neuroimaging, the NeuroSynth Project has mined fMRI-based brain activation maps from published x,y,z coordinate data tables from thousands of neuroimaging publications (Yarkoni et al., 2011). These literature-based methods can be contrasted with projects such as NeuroMorpho.org (Parekh and Ascoli, 2013) and ModelDB (Migliore et al., 2003; Hines et al., 2004), which index neuron morphological reconstructions and computational models for simulating neuron activity by obtaining this information directly from investigators.

Success among these projects can be defined according to different criteria. Such criteria include completeness and comprehensiveness; for example, what percentage of relevant connectivity studies are indexed within BAMS? How many different neuron types are contained within the NeuroMorpho database? Alternatively, success can be defined in terms of the utility of these databases in driving subsequent research, like the use of BAMS as a resource for discovering relationships between brain region connectivity and gene expression (French and Pavlidis, 2011) or the use of NeuroMorpho to discover general scaling relationships among the morphologies of neuron types (Teeter and Stevens, 2011). Similarly, NeuroSynth is widely used by cognitive scientists as a starting point for designing functional imaging studies. Thus while these projects are not yet comprehensive and likely contain data records of varying quality, these resources may nevertheless be employed to draw novel inferences.

These projects are logically divided according to their methods for obtaining the source data: through the use of manual methods like expert curation or user contributions versus automated methods such as text-mining. Notably, these approaches differ in their scale and accuracy; while algorithmic methods can "scale-up" and be applied to arbitrary numbers of publications, they typically have a lower accuracy relative to human-curated content (French et al., 2009). This lower accuracy is often attributed to the rich lexical complexity of biomedical texts which often require considerable context and background knowledge to understand and parse (Dickman, 2003; Ambert and Cohen, 2012). The competing constraints of scale versus accuracy pose a challenge for large-scale compilation of neuroscientific data.

Here, we built a custom infrastructure framework for extracting electrophysiological measurements for specific neuron types from published neurophysiology articles. These measurements included properties such as input resistance and resting membrane potential, as well as associated metadata (i.e., article-specific methodological details). Our methods combine algorithmic literature text-mining, drawing from the approach used by NeuroSynth (Yarkoni et al., 2011) where neurophysiological measurements are primarily extracted from data tables, as well as manual curation, leveraging the background knowledge of domain experts. The resulting neurophysiology database, named NeuroElectro, can be interactively viewed and explored through a public web interface at http://neuroelectro.org.

2. MATERIALS, METHODS, AND RESULTS

2.1. OVERVIEW

We describe and validate our semi-automated methodology for obtaining neuronal biophysical measurements directly from published reports in the literature (summarized in **Figure 1**). After obtaining full article texts from publishers, we then used text-mining algorithms to identify concepts specific to electrophysiology and neuron types, which we then validated manually.

2.2. ARTICLE IDENTIFICATION

We obtained electrophysiological data from 10 neuroscience specific journals (**Table 1**), which include: *Journal of Neuroscience, Journal of Neurophysiology,* and *Journal of Physiology* (among others). We selected these journals because they often devote a significant fraction of an article's main text, tables, and figures to detailed characterizations and summaries of intrinsic neuronal biophysical properties.

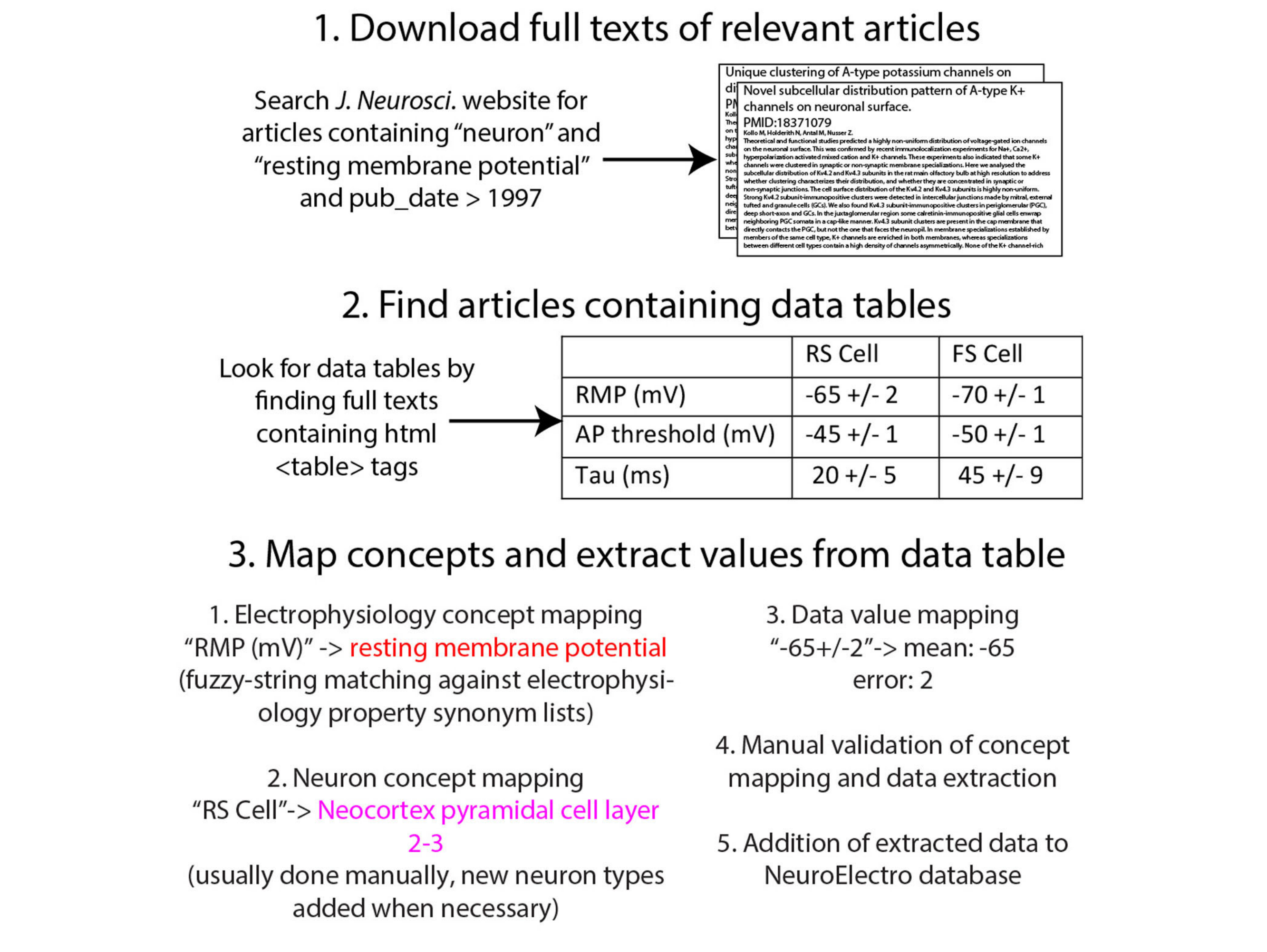
We obtained tens of thousands of potentially relevant full article texts directly from publisher websites. We first identified potential articles that were likely to contain information relevant to neuron biophysics using the native search functions provided within the journal websites and only downloaded articles containing in their full text any of a specific list of terms including "input resistance" and "resting membrane potential" (**Figure 1**). This pre-selection step allowed us to identify and download only articles that contained data relevant to our project. Upon identifying candidate articles, we then downloaded the full text of each potentially-relevant article as HTML; articles downloaded from the publisher Elsevier (e.g., *Neuron* and *Brain Research*) were downloaded as XML using the provided text-mining API and subsequently converted to HTML. We chose to work with HTML (as opposed to PDF or XML) because HTML provides a machine-readable markup of the article's content, allowing us easily to identify relevant elements within the article—such as data tables and the Methods section—using publicly available HTML-parsing tools (here we used the Beautiful Soup HTML-processing library implemented in Python: http://www.crummy.com/software/BeautifulSoup/bs4/doc/). Furthermore, because HTML is a single semi-structured standard used across publishers, we could write relatively generic HTML-processing algorithms applicable to content published across journals. Our focus on using HTML limits us to relatively newer articles—typically those published after 1996—because before this time most publications are only available as scanned PDF files. However, because the rate of publication across the field has grown exponentially, this HTML-available subset constitutes the majority of published neuroscience articles.

We stored the HTML-enhanced full text of each article in our database and associated each article with its corresponding PubMed ID (http://www.ncbi.nlm.nih.gov). These 8-digit IDs serve as publisher-independent unique identifiers for each article, and allow us to use PubMed-specific tools, such as a powerful API (i.e., PubMed eutils, http://www.ncbi.nlm.nih.gov/books/NBK25500/). For example, this API provides the ability to query each article's MeSH terms (MEdical Subject Headings) and returns basic methodological information such as animal species and strain.

2.3. ELECTROPHYSIOLOGICAL PROPERTY IDENTIFICATION

2.3.1. Rationale for focusing on electrophysiological property extraction from data tables

In order to algorithmically extract information on neuron electrophysiology from these articles, we needed to first specify the data types of interest. Our preference was to obtain as much detailed information about neuron electrophysiological properties as possible: ideally, this would include raw data corresponding to recorded electrophysiological traces. In mining information from articles, we were presented with multiple options (illustrated in **Figure 2**), including extraction from: (1) the text of the article including figure captions, (2) the figures of the article, or (3) data tables presented within the article. In addition to these, authors often submit supplemental materials and figures which also contain neurophysiological data.



1. Download full texts of relevant articles

Search *J. Neurosci.* website for articles containing "neuron" and "resting membrane potential" and pub_date > 1997

Unique clustering of A-type potassium channels on

Novel subcellular distribution pattern of A-type K+ channels on neuronal surface.
PMID:18371079
Kollo M, Holderith N, Antal M, Nusser Z.
Theoretical and functional studies predicted a highly non-uniform distribution of voltage-gated ion channels on the neuronal surface. This was confirmed by recent immunolocalization experiments for Na+, Ca2+, hyperpolarization activated mixed cation and K+ channels. These experiments also indicated that some K+ channels were clustered in synaptic or non-synaptic membrane specializations. Here we analysed the subcellular distribution of Kv4.2 and Kv4.3 subunits in the rat main olfactory bulb at high resolution to address whether clustering characterizes their distribution, and whether they are concentrated in synaptic or non-synaptic junctions. The cell surface distribution of the Kv4.2 and Kv4.3 subunits is highly non-uniform. Strong Kv4.2 subunit-immunopositive clusters were detected in intercellular junctions made by mitral, external tufted and granule cells (GCs). We also found Kv4.3 subunit-immunopositive clusters in periglomerular (PGC), deep short-axon and GCs. In the juxtaglomerular region some calretinin-immunopositive glial cells enwrap neighboring PGC somata in a cap-like manner. Kv4.3 subunit clusters are present in the cap membrane that directly contacts the PGC, but not the one that faces the neuropil. In membrane specializations established by members of the same cell type, K+ channels are enriched in both membranes, whereas specializations between different cell types contain a high density of channels asymmetrically. None of the K+ channel-rich

2. Find articles containing data tables

Look for data tables by finding full texts containing html <table> tags

	RS Cell	FS Cell
RMP (mV)	-65 +/- 2	-70 +/- 1
AP threshold (mV)	-45 +/- 1	-50 +/- 1
Tau (ms)	20 +/- 5	45 +/- 9

3. Map concepts and extract values from data table

1. Electrophysiology concept mapping
"RMP (mV)" -> resting membrane potential
(fuzzy-string matching against electrophysiology property synonym lists)

2. Neuron concept mapping
"RS Cell"-> Neocortex pyramidal cell layer 2-3
(usually done manually, new neuron types added when necessary)

3. Data value mapping
"-65+/-2"-> mean: -65
error: 2

4. Manual validation of concept mapping and data extraction

5. Addition of extracted data to NeuroElectro database

FIGURE 1 | Illustration of workflow for obtaining electrophysiological information from the research literature.

Given the challenges in mining raw electrophysiological traces from figure images, we instead focused on obtaining information about basic neuronal electrophysiological properties, such as input resistances and resting membrane potentials. Though this information is often presented within the text of the article, it is usually presented in complex sentence structures that are difficult to accurately parse algorithmically. Published data tables, on the other hand, present a unique opportunity for electrophysiological data extraction, since common techniques exist for extracting information from structured tables (Yarkoni et al., 2011). Moreover, because tables succinctly summarize multiple attributes of a collected dataset, the effort of an expert curator can be put to best use when validating tables relative to validating content mined from article sentences or figure panels. While we estimate that only 5–10% of electrophysiology articles contain data tables, there is sufficient redundancy within the field (i.e., multiple investigators often publish articles on the same neuron type) that focusing on data tables nevertheless yields substantial coverage of electrophysiological properties across many major neuron types.

2.3.2. Extracting information on electrophysiological properties

In extracting electrophysiological data, we took advantage of the fact that certain measurements are commonly made during intracellular recordings. For example, such recordings are commonly used to: (1) measure a neuron's resting membrane potential, (2) apply hyperpolarizing current injections for measurement of input resistance and membrane time constant, and (3) apply depolarizing current steps to evoke action potentials (spikes) and enable measurement of characteristics such as spike threshold, width, and amplitude.

We developed an electrophysiological lexicon comprising 28 measurements that we found to be commonly reported in the literature, largely based on previously published definitions (Toledo-Rodriguez et al., 2004; Ascoli et al., 2008). To account for subtle differences in terminology that authors use to refer to the same electrophysiological concept (e.g., resting membrane potential is often referred to as "rmp" and "V_{rest}"), we also identified a common list of synonyms to map to each concept. Together, these electrophysiological concepts and their synonyms define a preliminary ontology for electrophysiological concepts (included in

Table 1 | Statistics of journals represented in the NeuroElectro database.

Journal	Articles obtained	Validated	Not validated
J. Neurosci.	19,002	104	560
J. Neurophysiol.	12,078	94	555
J. Physiol. (Lond.)	10,543	44	235
Neuroscience	3035	14	205
Eur. J. Neurosci.	2495	7	117
Brain Res.	3017	7	146
Neuron	1657	4	43
Epilepsia	463	2	23
Neurosci. Lett.	1468	2	34
Hippocampus	208	2	10

Listing of journals and counts of articles downloaded (articles obtained), articles with published data tables containing neurophysiological information which has been manually validated by an expert curator (validated), and articles which likely contain information in a data table which has not yet been manually curated (not validated). Not validated articles are those which have at least four algorithmically assigned electrophysiological concepts within data tables.

Supplemental Materials). Moreover, this physiological measurement ontology can serve as a scaffolding for a more in-depth ontology of electrophysiological investigations (e.g., Ontology for Experimental Neurophysiology, Bruha et al., 2013). The terms in our preliminary ontology are also indexed and defined within NeuroLex (http://neurolex.org, Larson and Martone, 2013).

To identify data corresponding to electrophysiological properties reported within a data table, we developed algorithms to search data table header elements and assess whether these elements corresponded to any of the electrophysiological concept synonyms in our ontology. We first identified table header elements by searching for table elements composed primarily of non-numeric characters. For each putative header element, we then used fuzzy string matching algorithms (implemented using the fuzzywuzzy library in Python: https://github.com/seatgeek/fuzzywuzzy), to assess the textual match between the header element and each of the electrophysiological synonyms. These fuzzy matching algorithms combine a number of string match metrics into a single "match value," including whether a pair of strings completely match, contain matching substrings, or contain matching but misordered substrings. If the table header and electrophysiological synonym match value exceeded a specified threshold, the table header and corresponding row or column of numeric values were automatically mapped to the electrophysiological concept. Similarly, we mapped whole rows or columns to specific neuron types recorded during normotypic or "wild-type" conditions.

We then manually corrected cases where these algorithms mis-assigned an electrophysiological concept. For example, a common algorithmic mis-assignment was the case when an author used the string "EPSP amplitude" to refer to the electrophysiological concept excitatory post-synaptic potential amplitude. In these cases, our algorithms incorrectly mapped this string to "spike amplitude" because the former concept is not in our current ontology. In a test sample of 279 articles that were manually curated, we found that 78% of concept-matchings (901/1152) were identified correctly with no supervision, with the remainder manually corrected.

2.3.3. *Accounting for differences in electrophysiological definitions across investigators*

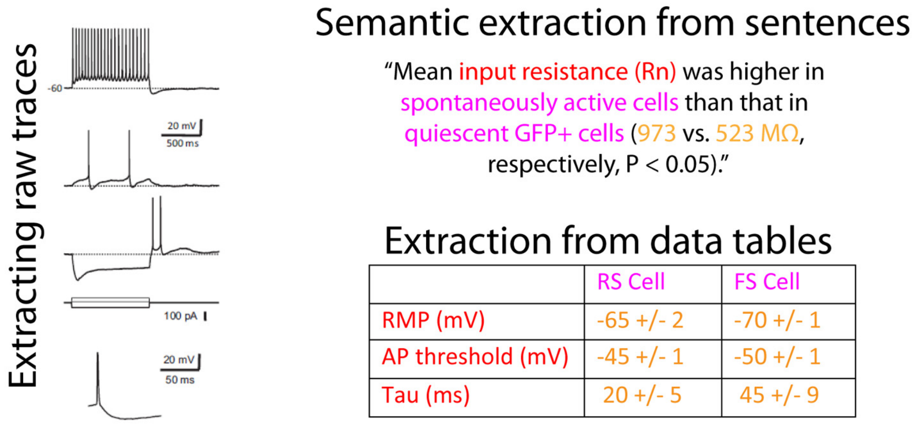
By focusing on textually matching the electrophysiological terms in each table to a list of electrophysiological concepts, we are implicitly assuming that electrophysiological properties are measured in the same way by investigators across different articles. For example, the most common method that electrophysiologists use to measure a neuron's spike properties is to record from the neuron in current-clamp mode and apply peri-threshold depolarizing currents to evoke 1–2 spikes over several hundred milliseconds or more. The neuron's spike amplitude is then commonly measured by calculating the difference between the neuron's voltage at spike threshold and spike peak for the first evoked spike (e.g., Connors et al., 1982; Toledo-Rodriguez et al., 2004). However, experimental differences exist between how investigators measure and compute these properties; we divide these differences into roughly three categories: *protocol*, *calculation*, and *condition* differences. For example, investigators can use different experimental protocols to measure the spike amplitude, like evoking spikes using current steps much greater than rheobase current required to elicit a single spike (*protocol differences*). Additionally, the spike amplitude itself can be calculated in different ways, such as using the neuron's resting membrane potential as the baseline instead of the spike threshold (*calculation differences*). Furthermore, the value of spike amplitude that an investigator reports will also be affected by specific experimental conditions such as the animal species or age and recording solution temperature or contents (*condition differences*).

When manually curating the text-mined content for some of the most commonly reported electrophysiological properties, we accounted for an investigator's calculation of an electrophysiological measurement using an inconsistent methodology (e.g., protocol or calculation differences). We did so by normalizing such measurements to a common reference definition or removing such data when normalization was not possible. However, we note that we could not identify all of these cases (in particular: spike amplitude, input resistance, and membrane time constant), in part because investigators did not always explicitly define how these measurements were calculated within their article. We note that in cases where we pool measurements which are measured using inconsistent protocols or calculations, this will tend to add unexplained variance to our data set. Given these measurement inconsistencies, we provide our recommendations for how these electrophysiological properties should be reported in future investigations via our electrophysiology ontology (see Supplemental Materials).

2.4. NEURON TYPE IDENTIFICATION

2.4.1. *Using neuron types defined by NeuroLex*

To extract physiological information specific to individual neuron types, we had to identify which neuron types were reported in each article. However, in many cases uniquely identifying the neuron type(s) reported in any given study and mapping these



Semantic extraction from sentences

"Mean input resistance (Rn) was higher in spontaneously active cells than that in quiescent GFP+ cells (973 vs. 523 MΩ, respectively, P < 0.05)."

Extraction from data tables

	RS Cell	FS Cell
RMP (mV)	-65 +/- 2	-70 +/- 1
AP threshold (mV)	-45 +/- 1	-50 +/- 1
Tau (ms)	20 +/- 5	45 +/- 9

FIGURE 2 | Illustration of the sources within an article containing information relevant to neuron electrophysiological properties. Data on neuronal electrophysiological properties are presented within article figures and raw traces, sentences within the article text, and formatted data tables. The raw traces and example sentence are from van Brederode et al. (2011) and are reproduced with permission from The American Physiological Society and the data table is a constructed example. Colored text indicates electrophysiological concepts (red), neuron concepts (pink), or neurophysiological data (yellow).

to a canonical "neuron type" is difficult. This difficulty arises in part because investigators use different criteria for classifying neurons, including electrophysiological, morphological, or molecular characteristics (Ascoli et al., 2008; Fishell and Heintz, 2013; Huang and Zeng, 2013).

To define canonical neuron types, we chose to use an existing list of approximately 250 neuron types and definitions provided by NeuroLex, a community-sourced, expert-defined collection of neuron types (http://neurolex.org; Shepherd, 2003; Hamilton et al., 2012; Larson and Martone, 2013). Moreover, we chose to use NeuroLex to keep our database consistent with existing resources and to enable future researchers to combine these resources seamlessly. NeuroLex also provides synonyms for each neuron type, which we utilized to identify the neuron type(s) in each article. In cases where a neuron type was investigated in the literature across multiple articles but not indexed within NeuroLex (e.g., cerebellar nucleus neurons), we manually added this neuron type to our database's listing and provided this neuron type to the NeuroLex neuron curators for incorporation (Gordon Shepherd, personal communication). Our specific criteria for identifying each of the neuron types reflected in the database are given in the Supplemental Materials.

2.4.2. Identifying specific neuron types within an article

Because of the complexity in unambiguously identifying neuron types, we used a mixed text-mining and manual approach to map the neuron types studied in each article to canonical NeuroLex neuron types. First, we used text-mining algorithms to provide an initial "best guess" of the most likely neuron type. Specifically, we used a bag-of-words approach (Aldous, 1985) on the full article text. This approach ignores the serial structure of the words in the document and utilizes only the frequency of occurrence of each word within the document. We next compared the article's word-frequency histogram to the listing of neuron synonyms provided by NeuroLex, ranking all neuron types by their likelihood of being actually studied within that article. In comparison to articles that we manually curated, we found that this automated approach accurately identified the neurons studied in each article with an accuracy of 30% (120 of 399 total) and up to 55% when defining success as the studied neuron appearing as one of the top three neuron types suggested by the bag-of-words method. Because of the relatively low accuracy of an automated-only approach, we added a manual curation step where a curator identified the recorded neuron type using HTML drop down menus enriched by the bag-of-words search (e.g., **Figure 4**). As previously described, we mapped individual data table elements and corresponding rows or columns to specific neuron types recorded under normotypic conditions. We note that currently we only identify data from normotypic or "control" neurons represented in tables, but plan to identify data from additional conditions in future work (e.g., from pharmacologically manipulated or genetically modified animals).

2.5. EXTRACTION OF ELECTROPHYSIOLOGICAL DATA VALUES

After identifying specific electrophysiological properties and neuron types reported in a data table (corresponding to row or column table headers), we then algorithmically extracted the data corresponding to the table intersection of these (**Figure 3**). We developed custom string regular expressions (Thompson, 1968) to parse the string corresponding to the numeric data. Specifically, we found that data strings were often of the form: "XX ± YY (ZZ)," where XX, YY, and ZZ refer to the mean, error term, and sample size (i.e., the "n"), respectively. Often, the number of replicates or error measurement were not reported or were reported in alternative ways within the table. Presently, the error term is not resolved as either a standard deviation or standard error measurement in the current version of NeuroElectro, but could easily be resolved in future iterations.

When designing our processing algorithms, we parsed data strings from right to left: first searching for data entities contained within parentheses, then for entities contained to the right

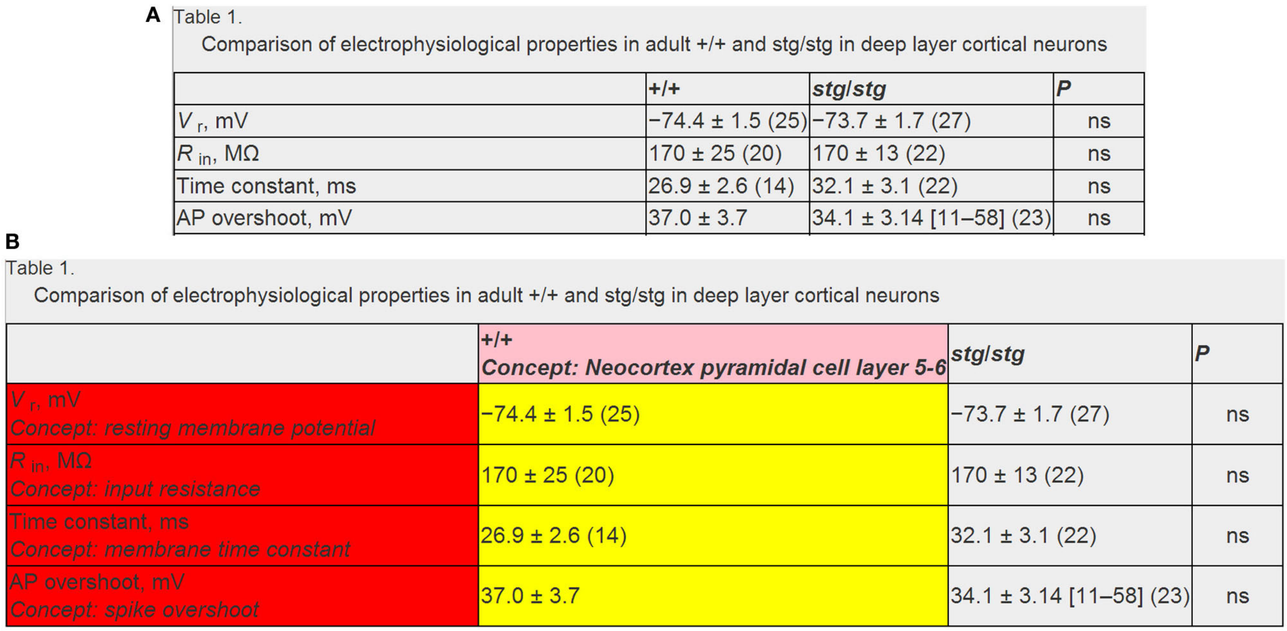


Table 1.
Comparison of electrophysiological properties in adult +/+ and stg/stg in deep layer cortical neurons

	+/+	*stg/stg*	*P*
V_{r}, mV	−74.4 ± 1.5 (25)	−73.7 ± 1.7 (27)	ns
R_{in}, MΩ	170 ± 25 (20)	170 ± 13 (22)	ns
Time constant, ms	26.9 ± 2.6 (14)	32.1 ± 3.1 (22)	ns
AP overshoot, mV	37.0 ± 3.7	34.1 ± 3.14 [11–58] (23)	ns

Table 1.
Comparison of electrophysiological properties in adult +/+ and stg/stg in deep layer cortical neurons

	+/+ *Concept: Neocortex pyramidal cell layer 5-6*	*stg/stg*	*P*
V_{r}, mV *Concept: resting membrane potential*	−74.4 ± 1.5 (25)	−73.7 ± 1.7 (27)	ns
R_{in}, MΩ *Concept: input resistance*	170 ± 25 (20)	170 ± 13 (22)	ns
Time constant, ms *Concept: membrane time constant*	26.9 ± 2.6 (14)	32.1 ± 3.1 (22)	ns
AP overshoot, mV *Concept: spike overshoot*	37.0 ± 3.7	34.1 ± 3.14 [11–58] (23)	ns

FIGURE 3 | Example data table illustrating mark-up and annotation of entities. (A) Example published data table containing neurophysiological information. Data table from Pasquale et al. (1997) and is reproduced with permission from The American Physiological Society. **(B)** Same as **(A)**, but semantically marked up with algorithmic and manually curated annotations. Markups in red and pink indicate electrophysiological and neuron type concepts and yellow indicates extracted data measurements. Note that here the textual string "+/+" and "stg/stg" refers to the normotypic and manipulated condition, respectively. Panels **(A)** and **(B)** reflect screenshots taken from NeuroElectro web interface.

of the ± term, and finally the remaining term which we assumed to refer to the mean term. We found that occasionally data were reported as "XX (LL–HH)"—where LL and HH indicate the lower and upper limits of a data range—and accounted for these cases similarly. We used regular expressions to identify entities such as digits, decimal signs, parentheses, and ± signs. We then converted the individual data elements which were encoded as textual strings of digits to double precision decimal entities before storing these into our database. Our focus here was primarily on parsing the mean value from a data record (i.e., summarizing the properties of a number of recorded neurons), but we also extracted and stored the error term and sample size where possible. Using these methods, we were able to extract 2176 electrophysiological values for 93 distinct neuron types within 279 articles.

2.6. MANUAL VALIDATION OF AUTOMATED DATA EXTRACTION

Following these automated concept identification and data extraction steps, we manually validated associated concepts and corrected incorrect concept mappings as necessary. We developed custom-HTML and javascript code to allow human curators to graphically interact with downloaded HTML data tables and "mark-up" entities within the table (**Figure 4**). This code allows for textual based elements of the HTML table to be semantically annotated using drop down menus and text fields. Moreover, because annotation is implemented via user interfaces composed of interactive web pages and drop down menus, these user interfaces are simple enough to be utilized by other expert curators with little formal instruction.

2.7. METADATA IDENTIFICATION

Given the strong relationships between experimental conditions, such as animal species or recording temperature, and electrophysiological measurements [e.g., input resistances are known to decrease when measured in neurons from older animals (Zhu, 2000; Okaty et al., 2009; Kinnischtzke et al., 2012)], we also identified information on article-specific experimental conditions by extracting this information primarily from each article's methods section. For each article, we found the methods section by developing custom HTML tag filters for each journal (e.g., common publisher-defined HTML tags for methods sections are "Methods" or "Experimental procedures"). For each metadata entity that we focused on (species, animal strain, electrode type, preparation type, liquid junction potential correction, animal age, recording temperature), we devised custom automated text searching methods to identify these based on combining regular expressions (Thompson, 1968) with PubMed MeSH terms (**Table 2**). In other words, rather than taking a machine-learning based approach and training classifiers (McCallum, 2002), we took a rule-based approach and developed custom rules for identifying metadata entities. For example, to identify whether the recording electrode's liquid junction potential was corrected for in the study (Neher, 1992), we searched for whether the character string "junction potential" was mentioned within the methods section and, if so, whether the sentence or phrase containing the term was explicitly negated (indicating that the junction potential was not corrected for). Here, we identified and parsed distinct sentences within the methods section using tools provided within the Natural Language Tool Kit in Python (Bird et al., 2009).

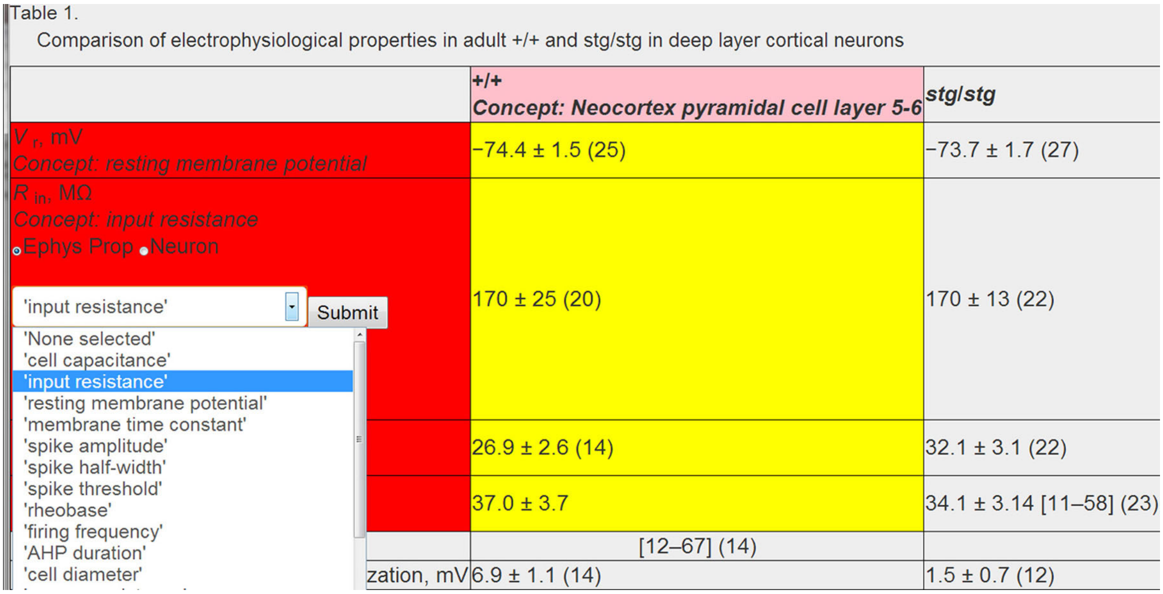


Table 1.
Comparison of electrophysiological properties in adult +/+ and stg/stg in deep layer cortical neurons

	+/+ *Concept: Neocortex pyramidal cell layer 5-6*	*stg/stg*
V r, mV *Concept: resting membrane potential*	−74.4 ± 1.5 (25)	−73.7 ± 1.7 (27)
R in, MΩ *Concept: input resistance* ● Ephys Prop ○ Neuron 'input resistance' Submit	170 ± 25 (20)	170 ± 13 (22)
	26.9 ± 2.6 (14)	32.1 ± 3.1 (22)
	37.0 ± 3.7	34.1 ± 3.14 [11–58] (23)
	[12–67] (14)	
zation, mV	6.9 ± 1.1 (14)	1.5 ± 0.7 (12)

FIGURE 4 | Example of human validation of algorithmically assigned content. All textual elements of a table are enhanced using HTML and javascript to allow for assignment of neuron or electrophysiological concepts using drop down menus. Example data table from Pasquale et al. (1997) and is reproduced with permission from The American Physiological Society.

Following automated identification of article metadata, we then manually checked each article to ascertain that algorithmically-tagged metadata was identified correctly and, as before, we corrected misidentified content as necessary through the use of custom HTML forms. We found that the mean accuracy of algorithmic metadata assignment was approximately 50% (**Figure 5**) and was typically lower for identifying continuous-valued metadata (e.g., animal age or recording temperature) relative to nominal metadata such as species and electrode type.

2.8. OBJECT MODELS AND RELATIONAL DATABASE

We stored extracted data and metadata using a relational database implemented in MySQL (http://dev.mysql.com/doc/refman/5.6/en/) built from a Python Django object model (https://www.djangoproject.com/). The object model contains classes for a number of fields, such as full article texts, electrophysiological properties, neuron types, synonyms, electrophysiological data values, and experimental metadata (**Figure S1**). A useful feature of the relational nature of the database is that it enables linking between classes (e.g., linking between neuron types and electrophysiological properties reported by a single investigator across multiple articles). This linking feature facilitates efficient and arbitrary querying of data; for example, querying for known electrophysiological data on olfactory bulb mitral cells recorded *in vitro* and published between the dates 2000 and 2004. For example, such a feature could be used to assess whether measurements of olfactory bulb mitral cells have changed as a function of time or are dependent upon whether the data are collected *in vitro* or *in vivo*.

2.9. WEB APPLICATION

The primary results of NeuroElectro are viewable at http://www.neuroelectro.org where the data can be interactively explored.

2.9.1. Human interface

The web interface is organized around neuron types and electrophysiological properties. For example, each neuron type has its own webpage where extracted data corresponding to specific electrophysiological properties is graphically and interactively displayed (graphical plot interactivity implemented using the jqPlot javascript toolbox, http://www.jqplot.com/). Users can thus visualize the mean and variability of electrophysiological values across papers, view references plus experimental metadata, and easily navigate to primary data from specific papers. Furthermore, users can view electrophysiological data across all of the neuron types in the database—putting phenotypic properties of a given neuron type into the larger context of other neuron types located throughout the nervous system.

The web application also contains preliminary features to allow website visitors to contribute to the NeuroElectro resource. For example, users can suggest articles that contain electrophysiological data which are not already in the database. We also invite visitors to become "expert curators" for neurons of interest. In the future, we plan to build functionality that will allow investigators to upload raw and summary data, such as recorded voltage and current traces. In addition, we plan to continue mining the literature and adding neurophysiological measurements as they are published.

2.9.2. API

An initial API (application programmer interface) providing public access to the electrophysiological data is described at http://neuroelectro.org/api/docs/. This RESTful API allows contents of the NeuroElectro database to be dynamically retrieved in JSON or XML format for utilization within external applications. For example, using the current API, a developer could build an application which dynamically queries NeuroElectro for all data

Table 2 | A partial listing of metadata attributes and extraction methodology.

Metadata concept	Values	Extraction method	Regular expression	MeSH term
Species		MeSH term only		
	Rats			Rats
	Mice			Mice
	Guinea pigs			Guinea pigs
Electrode type		MeSH term + Regex		
	Patch-clamp		"Whole cell" or "patch clamp"	Patch-clamp techniques
	sharp		"Sharp electrode"	
Animal strain		MeSH term only		
	Fischer 344			Rats, Inbred F344
	Long-evans			Rats, Long-Evans
	Sprague-Dawley			Rats, Sprague-Dawley
	Wistar			Rats, Wistar
	C57BL			Mice, Inbred C57BL
	BALB C			Mice, Inbred BALB C
Preparation type		MeSH Term + Regex		
	In vitro		"Slice" or "*in vitro*"	
	In vivo		"*In vivo*"	
	Cell culture		"Culture"	Cell culture techniques
	Model		"Model"	Computer simulation
Junction potential		Regex		
	Not corrected		"Not junction potential"	
	Corrected		"Junction potential"	
Recording temperature		Regex		
	Continuous value		"Record ... C" or "experiment C"	
	Room temperature		"Record room temperature"	
Animal age		Regex		
	Continuous value		Find digits near: "P#-#" or "P#-P#"	

Metadata attributes are extracted through combining PubMed Medical Subject Heading terms (MeSH Terms) and custom regular expressions (Regex). Regular expression column (or MeSH Term column) indicates specific regular expressions (or MeSH terms) used for identifying metadata concept entities.

corresponding to layer 2/3 neocortical pyramidal cells and then uses this data to constrain parameters for a Hodgkin–Huxley type neuron model (Hodgkin and Huxley, 1952). Example use cases of the current API (version 1) include:

- http://neuroelectro.org/api/1/n/ : Returns a list of all neurons with electrophysiological data indexed in NeuroElectro.
- http://neuroelectro.org/api/1/nedm/?nlex=sao830368389 : Returns a list of all indexed data on CA1 pyramidal cells (queried using the NeuroLex identifier for CA1 pyramidal cells, *sao830368389*).
- http://neuroelectro.org/api/1/nes/?e__name=Input+resistance: Returns a data record composed of the mean, standard deviation, and sample size n, summarizing input resistance measurements from cerebellar Purkinje cells based on all indexed articles in NeuroElectro database. Here the database query is performed using the textual strings for the electrophysiological and neuron type concepts.

Our future plans are to work with domain ontologists to further develop the existing API into a formal relational data format (RDF) specification, allowing further querying and extending of NeuroElectro into additional resources. All code used for the project is available at http://github.com/neuroelectro/neuroelectro.

3. DISCUSSION

We have developed, applied, and validated a methodology and pipeline for extracting—from existing literature on cellular neurophysiology—measurements of basic biophysical properties from diverse neuron types throughout the nervous system. Currently, the NeuroElectro database contains 2344 manually curated electrophysiological measurements from 98 neuron types from 335 publications. Of these electrophysiological measurements, 2176 (93%) were obtained from 279 (83%) publications using the semi-automated approach described here. In addition, we machine-extracted and manually validated 1667

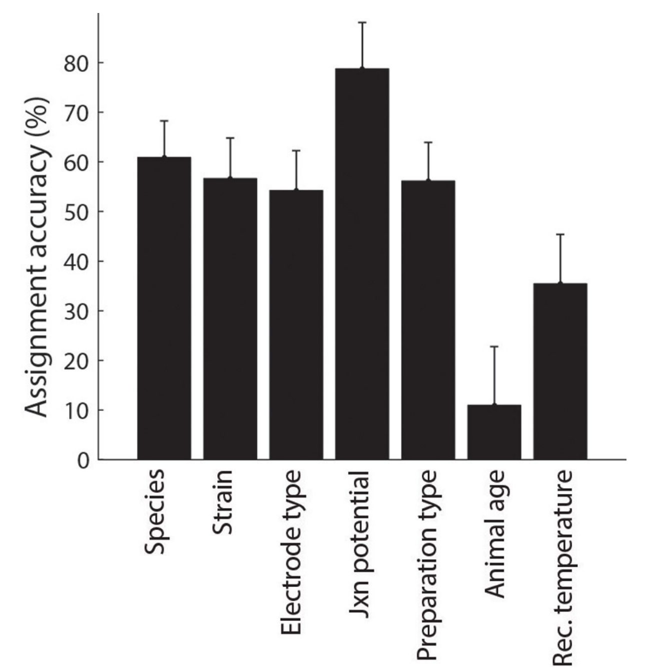


FIGURE 5 | Accuracy of metadata assignment using automated methods alone. Error indicate 95% binomial confidence intervals.

methodological conditions (metadata) from these publications. This represents the single largest collection of neurophysiological data ever compiled and represents a potentially valuable tool for scientific discovery.

3.1. SPECIFIC BENEFITS PROVIDED BY THE SEMI-AUTOMATED APPROACH

One of the key advantages of the approach described here is that the automated pipeline identifies publications which are likely to contain content relevant to our domain area (i.e., measurements of neuronal biophysics). Thus a human needs only to manually curate the content first identified by the algorithms as being likely relevant, instead of having to identify the relevant content *de novo*. Moreover, the automated identification of neuron types in articles allows us to target manual curation efforts to publications likely to contain data from specific neuron types, such as neurons that are currently underrepresented in the database.

Given our laboratory's focus on olfactory circuits, we conducted a natural experiment to compare the efficacy of biophysical property extraction using these semi-automated methods versus traditional methods which do not make use of algorithmic text-mining as a pre-processing step. In a seven-hour curation session (evoking the classic American parable of John Henry versus the steam-powered hammer), a senior graduate student in our laboratory identified 91 electrophysiological measurements (focusing on resting membrane potential, input resistance, membrane time constant, spike amplitude, spike width, and spike threshold) from 35 articles for 7 olfactory bulb neuron types using only prior knowledge of which articles and investigators were likely to have reported such electrophysiological data. In a comparable seven-hour curation session using our semi-automated methods, a single curator (with similar expertise to the first curator) identified 551 electrophysiological measurements from 70 articles across 40 neuron types throughout the nervous system. Moreover, this comparison would likely tilt even more in favor of the semi-automated methods had the curators been less familiar with the primary literature.

3.2. SCALABILITY OF CURRENT APPROACH

We note that multiple steps in our approach require manual intervention by an expert curator in order for electrophysiological measurements to be extracted with an acceptably low error rate. Namely, an expert curator needs to confirm which of the machine-identified candidate neuron types are recorded from in each article and where data from the normotypic or "control" states of these neurons are textually referenced within a data table. Moreover, given the current accuracy of the unsurpervised algorithmic assignment of electrophysiological concepts and experimental metadata (78% and 50%, respectively), these also need to be manually validated and corrected and normalized as required by an expert. Given the necessity of these manual steps, the scalability of our current approach is limited by our ability to manually curate this information or by our ability to improve the error rate of the automated methods. Despite this limitation, our current pipeline is still much faster than a purely manual one. The methodology could be further improved by correcting falsely matching entities (such as EPSP amplitude in section 2.3.2). These could be corrected by simply adding these valid concepts to the electrophysiolgical ontology. Moreover, these improvements would facilitate formally computing the sensitivity and specificity of these entity recognition methods.

3.3. PRELIMINARY USE OF NEUROELECTRO IN SCIENTIFIC WORK

The NeuroElectro project is intended to facilitate scientific investigation by providing easy access to large quantities of data about neurons. Because the data is machine-readable, we have already begun to conduct several analyses that would not be possible without this resource. First, we have begun an investigation of the relationships between neurons as defined by the similarity of their electrophysiological properties. This information can be used to make predictions about as yet unmeasured properties. Second, we have begun to explore the relationship between patterns of gene expression [using both the Allen Brain Atlas (Lein et al., 2007) and single cell qPCR approaches] and electrophysiological properties of neurons. Third, we have begun automated testing of quantitative neuron models in concert with SciUnit (Omar et al., 2014), under the reasonable assumption that these models should be constrained by the available experimental data. These projects are described in manuscripts currently in preparation.

3.4. EXTENSIONS AND IMPROVEMENTS TO THE CURRENT SEMI-AUTOMATED ALGORITHMS

Currently, neuron type identification is a critical bottleneck in our approach. One potential improvement would be to replace the non-specific bag-of-words approach we are currently using in favor of a bioNLP classifier-based approach (McCallum, 2002). Specifically, we propose adapting the named entity recognition methodology used by the WhiteText project for tagging brain regions mentioned in literature (French et al., 2009; French and Pavlidis, 2012) and first identifying spans of text likely to pertain to a neuron type before mapping these textual spans to a individual neuron type within the neuron ontology.

The approach described here is highly effective for extracting biophysical measurements presented within machine-readable data tables published within journal articles. However, the current requirement that these data tables exist in a machine parseable format, such as HTML or XML, limits this approach from being directly applied to older manuscripts, which are only available as scanned images. Existing approaches, such as optical character recognition technology (OCR; e.g., Ramakrishnan et al., 2012) may be applied toward this problem in the future.

Given the relatively low accuracy of the automated approach to identifying neuron types, there may be several avenues through which this process can be improved. For example, we note that the automated approach was particularly ineffective when the neuron type investigated within an article was not already described in NeuroLex or when the neuron had an insufficient list of synonyms associated with it. The current implementation of NeuroElectro also does not consider common neuron type acronyms (e.g., that olfactory bulb mitral cells are commonly referred to as "MCs"). Adding acronym and abbreviation identification to future iterations will thus likely improve the automated approach (Okazaki and Ananiadou, 2006; French and Pavlidis, 2012). Moreover, our current implementation of the bag-of-words algorithm would likely be enhanced via minor improvements, such as only identifying neurons using the text of the abstract or results and discarding text from the introduction or discussion. As neuron identification forms the major bottleneck in the scalability of NeuroElectro due to the requirement for manual curation, we plan to address this bottleneck in future revisions.

3.5. FUTURE METHODS FOR DATA EXTRACTION

A more pressing issue with the current approach is its focus on extraction from data tables. We estimate that only 5–10% of published electrophysiological data is contained within tables, while the remaining 90–95% is presented within article text or figure images. Given our preference to obtain data in their most raw form, we initially considered extraction of data from figures, e.g., voltage traces of neuronal activity. However, digitizing article figures (presented by publishers as images) into a form that can be further analyzed presents multiple challenges. Though techniques and tools exist to digitize figures, substantial amounts of manual effort are required to employ them correctly, making this figure-based approach difficult to scale to increasing numbers of articles without also employing a large team of human curators. While automatically extracting measurements from figure images will likely prove challenging, our methods can likely be adapted to operate on article text, perhaps by making use of bioNLP methodologies currently used for relationship extraction in the identification of connected brain regions (French et al., 2012) or interacting pairs of proteins (Kim and Wilbur, 2011).

Future developments in machine extraction of data from the scientific literature will be of great benefit. These should include better semantic understanding of context, ranging from relatively unambiguous notations such as units, to syntax-parsing of free-form prose that relates objects of study to their reported properties. Much progress has been made by computer scientists in some of these areas, and more future engagement with their research should enable vastly more data to be extracted from the literature.

We believe that, if successful, the use of NeuroElectro will influence the practices of scientists writing papers and reporting results. Specifically, we recommend the usage of common standards and definitions for basic physiological measurements (Toledo-Rodriguez et al., 2004) and neuron types (Ascoli et al., 2008; Larson and Martone, 2013). Moreover, we advocate that, where possible, scientists report more basic physiological data overall and report such data using machine-parsable data tables. These recommendations could be made informally by journals (in particular, requested by reviewers during manuscript review) as well as by funding agencies. This change would make it easier for scientists to find and make use of data collected by others. Such a culture shift has the potential to make science function more effectively and efficiently to facilitate discovery.

FUNDING

This work was supported by a National Science Foundation Graduate Research Fellowship and a R. K. Mellon Foundation Fellowship (to Shreejoy J. Tripathy), an Achievement Rewards for College Scientists Foundation Fellowship and NIDCD NRSA F31DC013490 (to Shawn D. Burton), NIDCD award F32DC010535 and NIMH award R01MH081905 (in support of Richard C. Gerkin), and NIDCD award R01DC005798 (to Nathaniel N. Urban).

ACKNOWLEDGMENTS

We thank William Cohen, Gordon Shepherd, and Renaud Richardet for discussions and comments on the manuscript. We are especially grateful to all of the investigators whose collected data are represented within the NeuroElectro database. We thank the academic journal publishers (in particular, Elsevier and Wiley and Highwire) for allowing us access to their full-texts for text-mining. We are especially grateful to all of the investigators whose collected data are represented within the NeuroElectro database.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/journal/10.3389/fninf.2014.00040/abstract

Figure S1 | Illustration of NeuroElectro relational database schema.

REFERENCES

Aldous, D. J. (1985). "Exchangeability and related topics," in *École d'été de Probabilités de Saint-Flour XIII 1983*. Lecture notes in mathematics, Vol. 1117, ed P. L. Hennequin (Berlin; Heidelberg: Springer), 1–198.

Ambert, K. H., and Cohen, A. M. (2012). Text-mining and neuroscience. *Int. Rev. Neurobiol.* 103, 109–132. doi: 10.1016/B978-0-12-388408-4.00006-X

Ascoli, G. A., Alonso-Nanclares, L., Anderson, S. A., Barrionuevo, G., Benavides-Piccione, R., Burkhalter, A., et al. (2008). Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex. *Nat. Rev. Neurosci.* 9, 557–568. doi: 10.1038/nrn2402

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. Beijing; Cambridge, MA: O'Reilly.

Bota, M., Dong, H.-W., and Swanson, L. W. (2005). Brain architecture management system. *Neuroinformatics* 3, 15–48. doi: 10.1385/NI:3:1:015

Bruha, P., Papez, V., Bandrowski, A., Grewe, J., Mouček, R., Tripathy, S., et al. (2013). The ontology for experimental neurophysiology: a first step toward semantic annotations of neurophysiology data and metadata. *Front. Neuroinform. (Conference Abstract: Neuroinformatics 2013)* 26. doi: 10.3389/conf.fninf.2013.09.00026

Connors, B. W., Gutnick, M. J., and Prince, D. A. (1982). Electrophysiological properties of neocortical neurons *in vitro*. *J. Neurophysiol.* 48, 1302–1320.

Dickman, S. (2003). Tough mining. *PLoS Biol.* 1:e48. doi: 10.1371/journal.pbio.0000048

Fishell, G., and Heintz, N. (2013). The neuron identity problem: form meets function. *Neuron* 80, 602–612. doi: 10.1016/j.neuron.2013.10.035

French, L., Lane, S., Xu, L., and Pavlidis, P. (2009). Automated recognition of brain region mentions in neuroscience literature. *Front. Neuroinform.* 3:29. doi: 10.3389/neuro.11.029.2009

French, L., Lane, S., Xu, L., Siu, C., Kwok, C., Chen, Y., et al. (2012). Application and evaluation of automated methods to extract neuroanatomical connectivity statements from free text. *Bioinformatics* 28, 2963–2970. doi: 10.1093/bioinformatics/bts542

French, L., and Pavlidis, P. (2011). Relationships between gene expression and brain wiring in the adult rodent brain. *PLoS Comput. Biol.* 7:e1001049. doi: 10.1371/journal.pcbi.1001049

French, L., and Pavlidis, P. (2012). Using text mining to link journal articles to neuroanatomical databases. *J. Comp. Neurol.* 520, 1772–1783. doi: 10.1002/cne.23012

Hamilton, D. J., Shepherd, G. M., Martone, M. E., and Ascoli, G. A. (2012). An ontological approach to describing neurons and their relationships. *Front. Neuroinform.* 6:15. doi: 10.3389/fninf.2012.00015

Hines, M. L., Morse, T., Migliore, M., Carnevale, N. T., and Shepherd, G. M. (2004). ModelDB: a database to support computational neuroscience. *J. Comput. Neurosci.* 17, 7–11. doi: 10.1023/B:JCNS.0000023869.22017.2e

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544.

Huang, J., and Zeng, H. (2013). Genetic approaches to neural circuits in the mouse. *Annu. Rev. Neurosci.* 36, 183–215. doi: 10.1146/annurev-neuro-062012-170307

Kim, S., and Wilbur, W. J. (2011). Classifying protein-protein interaction articles using word and syntactic features. *BMC Bioinform.* 12:S9. doi: 10.1186/1471-2105-12-S8-S9

Kinnischtzke, A. K., Sewall, A. M., Berkepile, J. M., and Fanselow, E. E. (2012). Postnatal maturation of somatostatin-expressing inhibitory cells in the somatosensory cortex of GIN mice. *Front. Neural Circ.* 6:33. doi: 10.3389/fncir.2012.00033

Larson, S. D., and Martone, M. E. (2013). NeuroLex.org: an online framework for neuroscience knowledge. *Front. Neuroinform.* 7:18. doi: 10.3389/fninf.2013.00018

Lein, E. S., Hawrylycz, M. J., Ao, N., Ayres, M., Bensinger, A., Bernard, A., et al. (2007). Genome-wide atlas of gene expression in the adult mouse brain. *Nature* 445, 168–176. doi: 10.1038/nature05453

McCallum, A. K. (2002). *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu

Migliore, M., Morse, T. M., Davison, A. P., Marenco, L., Shepherd, G. M., and Hines, M. L. (2003). ModelDB. *Neuroinformatics* 1, 135–139. doi: 10.1385/NI:1:1:135

Neher, E. (1992). "Correction for liquid junction potentials in patch clamp experiments," in *Methods in Enzymology*, Vol. 207, ed B. Rudy (New York, NY: Academic Press), 123–131.

Okaty, B. W., Miller, M. N., Sugino, K., Hempel, C. M., and Nelson, S. B. (2009). Transcriptional and electrophysiological maturation of neocortical fast-spiking GABAergic interneurons. *J. Neurosci.* 29, 7040–7052. doi: 10.1523/JNEUROSCI.0105-09.2009

Okazaki, N., and Ananiadou, S. (2006). Building an abbreviation dictionary using a term recognition approach. *Bioinformatics* 22, 3089–3095. doi: 10.1093/bioinformatics/btl534

Omar, C., Aldrich, J., and Gerkin, R. (2014). "Collaborative infrastructure for testdriven scientific model validation," in *Proceedings of the 36th International Conference on Software Engineering, ICSE '14* (to appear), (Hyderabad: ACM). Available online at: https://github.com/cyrus-/papers/blob/master/sciunit-icse14/sciunit-icse14.pdf?raw=true

Parekh, R., and Ascoli, G. (2013). Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron* 77, 1017–1038. doi: 10.1016/j.neuron.2013.03.008

Pasquale, E. D., Keegan, K. D., and Noebels, J. L. (1997). Increased excitability and inward rectification in layer v cortical pyramidal neurons in the epileptic mutant mouse stargazer. *J. Neurophysiol.* 77, 621–631.

Ramakrishnan, C., Patnia, A., Hovy, E., and Burns, G. A. (2012). Layout-aware text extraction from full-text PDF of scientific articles. *Source Code Biol. Med.* 7:7. doi: 10.1186/1751-0473-7-7

Shepherd, G. M. (ed.) (2003). *The Synaptic Organization of the Brain, 5th Edn*. Oxford, NY: Oxford University Press.

Teeter, C., and Stevens, C. (2011). A general principle of neural arbor branch density. *Curr. Biol.* 21, 2105–2108. doi: 10.1016/j.cub.2011.11.013

Thompson, K. (1968). Programming techniques: regular expression search algorithm. *Commun. ACM* 11, 419–422. doi: 10.1145/363347.363387

Toledo-Rodriguez, M., Blumenfeld, B., Wu, C., Luo, J., Attali, B., Goodman, P., et al. (2004). Correlation maps allow neuronal electrical properties to be predicted from single-cell gene expression profiles in rat neocortex. *Cereb. Cortex* 14, 1310–1327. doi: 10.1093/cercor/bhh092

van Brederode, J. F. M., Yanagawa, Y., and Berger, A. J. (2011). GAD67-GFP+ neurons in the nucleus of roller: a possible source of inhibitory input to hypoglossal motoneurons. I. Morphology and firing properties. *J. Neurophysiol.* 105, 235–248. doi: 10.1152/jn.00493.2010

Yarkoni, T., Poldrack, R. A., Nichols, T. E., Van Essen, D. C., and Wager, T. D. (2011). Large-scale automated synthesis of human functional neuroimaging data. *Nat. Methods* 8, 665–670. doi: 10.1038/nmeth.1635

Zhu, J. J. (2000). Maturation of layer 5 neocortical pyramidal neurons: amplifying salient layer 1 and layer 4 inputs by ca^{2+} action potentials in adult rat tuft dendrites. *J. Physiol.* 526, 571–587. doi: 10.1111/j.1469-7793.2000.00571.x

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 02 December 2013; accepted: 27 March 2014; published online: 29 April 2014.

Citation: Tripathy SJ, Savitskaya J, Burton SD, Urban NN and Gerkin RC (2014) NeuroElectro: a window to the world's neuron electrophysiology data. Front. Neuroinform. ***8****:40. doi: 10.3389/fninf.2014.00040*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 27 May 2014
doi: 10.3389/fninf.2014.00053

Web based tools for visualizing imaging data and development of XNATView, a zero footprint image viewer

David A. Gutman[1]*, William D. Dunn Jr[1], Jake Cobb[2], Richard M. Stoner[3], Jayashree Kalpathy-Cramer[4] and Bradley Erickson[5]

[1] *Department of Biomedical Informatics, Emory University, Atlanta, GA, USA*
[2] *Georgia Institute of Technology, College of Computing, Atlanta, GA, USA*
[3] *Department of Neurosciences, University of California San Diego School of Medicine, La Jolla, CA, USA*
[4] *Harvard-MIT Division of Health Sciences and Technology, Martinos Center for Biomedical Imaging, Charlestown, MA, USA*
[5] *Department of Radiology, Mayo Clinic, Rochester, MN, USA*

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Mingrui Xia, Beijing Normal University, China
Ingvar Christian Lagerstedt, European Microbiology Laboratory - European Bioinformatics Institute, UK

****Correspondence:***
David A. Gutman, Department of Biomedical Informatics, Emory University, 36 Eagle Row, Room 572 PAIS, Atlanta, GA 30322, USA
e-mail: dgutman@emory.edu

Advances in web technologies now allow direct visualization of imaging data sets without necessitating the download of large file sets or the installation of software. This allows centralization of file storage and facilitates image review and analysis. XNATView is a light framework recently developed in our lab to visualize DICOM images stored in The Extensible Neuroimaging Archive Toolkit (XNAT). It consists of a PyXNAT-based framework to wrap around the REST application programming interface (API) and query the data in XNAT. XNATView was developed to simplify quality assurance, help organize imaging data, and facilitate data sharing for intra- and inter-laboratory collaborations. Its zero-footprint design allows the user to connect to XNAT from a web browser, navigate through projects, experiments, and subjects, and view DICOM images with accompanying metadata all within a single viewing instance.

Keywords: radiology, MRI, DICOM-viewer, XNAT, web-based image viewer, PyXNAT, biomedical imaging

INTRODUCTION

Data management challenges regularly pose problems among imaging laboratories. Visualization and sharing of complex imaging data sets has traditionally involved downloading large file sets, installing custom software applications, or in some cases simply sharing screenshots of specific images with colleagues for review and comment. These inefficient *ad hoc* solutions often make collaboration and soliciting feedback on imaging data complicated, imprecise, and time intensive. However, recent advances in web-based technologies such as HTML5 and faster overall internet connectivity have the potential to significantly simplify this process.

In this work, we review some of the existing source tools and libraries that facilitate web-based image visualization of imaging data sets. While DICOM is the *lingua franca* of clinical imaging, it is worth noting many other imaging formats are commonly used in imaging research. We will therefore also review some tools and frameworks that support imaging formats in addition to DICOM.

A number of emerging technologies allow visualization and interaction with not only the image itself, but also with derivative images such as masks, tractography results, and statistical maps. We will review various tools currently available that support this functionality offline as well as demonstrate our current work that allows visualization of image overlays directly via HTML.

Capitalizing on these recent advances, we have developed a light weight HTML based image browser that integrates XNAT, a popular research informatics platform which we will describe later. The need for such an image-viewer stemmed from an ongoing project in our lab which involved the organization and curation of large retrospectively-collected imaging data sets of cancer patients with high grade gliomas. For our initial project (Gutman et al., 2013), we were presented with hundreds of volumes of MR imaging sets and we needed an efficient method to select which MRI cases were appropriate for the study. Specifically, we wanted to pull patients with pre-surgical/pre-treatment T2 FLAIR as well as both pre and post gadolinium contrast T1 sequences with sufficient image quality. Unlike typical neuroimaging studies where data is collected on a single MRI machine with a well-defined imaging protocol, the imaging data in this case was collected during a period of more than a decade from various universities within the TCGA network (Cancer Genome Atlas Research et al., 2013), oftentimes using different image scanners and following different protocols.

Due to the heterogeneity of imaging protocols in the clinical setting, our data was interspersed with DICOM images of insufficient quality (motion artifacts, limited field of view, missing slices) as well as ambiguous/improper names (i.e., we had found that several T1 images had been incorrectly labeled as T2). This task was further confounded by the necessities of anonymization, as well as by a lack of a direct link back to clinical data that would have indicated the treatment status of the patient at a given scan (e.g., pre/post-surgery). Additional complications included numerous seemingly duplicate/extraneous scans (e.g., two consecutive images labeled "T2 FLAIR," or three different images all labeled "T1 Gad") which needed to be disambiguated. In a clinical environment, a scan technician may repeat a scan due to poor image quality without needing a way to label the "good" scan. As this data is of course not available many years later when the data is being analyzed for research purposes, the ability to rapidly open and compare images is thus critical. Due to the massive volume of patient data, we therefore wanted some type of quality assurance tool that we could use to browse through the images to quickly determine if the corresponding labels and metadata were correct.

While a built-in image viewer is available through XNAT, the time required to select and to view individual scans from patient to patient made this option too inefficient for our project. In addition, the viewer uses Java, which presented a number of practical challenges, where programs can not readily be updated (e.g., Java) on university- or hospital-owned equipment. To address these issues, we developed XNATView, a tool that allows us and potentially any lab involved in large population neuroradiological research to easily review large sets of image sequences solely from a web browser.

While currently supporting direct integration with XNAT, the XNATView interface can be easily modified to communicate with any service that supports query and retrieval of DICOM images. As a proof of principle, we will present some of our prototype work integrating XNATView, through various plugins, with the Platform to Enable Shared Scientific Computing And Research Advances (PESSCARA) being developed at the Mayo Clinic. We have used the term zero-foot print viewer to describe software that does not require installation of additional software (e.g., Java, Active-X, Flash, etc.) relying on native functionality of the web browser.

BACKGROUND

RESEARCH IMAGE MANAGEMENT SYSTEMS/PACS

The basic technology to support the standardized sharing of medical images is a Picture Archiving and Communication System (PACS) (Bryan et al., 1999). The basic structure includes a secure one-way interface transmitting DICOM formatted images from the physical data capture (X-ray, CT, MRI, etc.), which are ideally (although optionally) transmitted to a quality assurance workstation (PACS gateway) where demographics and other characteristics are verified. The images are then transferred to a centralized archive, where they can be queried and viewed by radiologists in a reading workstation. Numerous vendors have developed their own PACS workstations, with varying capabilities ranging from simply browsing 2-D slices to allowing 3-D visualizations and advanced image reconstruction capabilities.

For the purposes of this review, we will limit ourselves to freely-available open-source based platforms. Resources such as http://idoimaging.com and http://nitrc.org list a number of available medical software packages as well as accompanying information of the tools.

Among the most versatile and earliest implementations of an open-source web-based viewer is Weasis, a program available through the DCM4CHE application collection. DCM4CHE is a DICOM archive and image manager that can be entirely run from a web browser (http://dcm4che.org/). It was developed within the framework of JDicom, a toolkit written in Java (Warnock et al., 2007) and is currently distributed by the developers of the DCMTK toolkit (DICOM@OFFIS, 2013). Personal correspondences have highlighted that DCM4CHE is especially appropriate for large databases, such as those on a university or hospital setting, and runs smoothly on Windows or Linux machines. Importantly, DCM4CHE offers various storage, clinical, and sharing features which were designed around contemporary standards such as HL7 and DICOM to facilitate interoperability between users. DCM4CHE can serve as an image source for DICOM compliant applications (such as ClearCanvas and OsiriX/etc.) as well as an integrated web-based image viewer through the Weasis application. Weasis is a multi-purpose clinical image viewer designed to view images stored in a PACS with minor adjustments (**Figure 1**). CDMedicPACSWeb provides (http://cdmedicpacsweb.sourceforge.net/CDMEDIC_PACS_WEB.html) a virtual machine/base installation which has WEASIS and DCM4CHEE preconfigured.

While these tools are useful when DICOM is the primary imaging modality, other more comprehensive systems that support multiple imaging formats have also been developed.

A popular PACS workstation program on the Macintosh Platform is OsiriX (OSIRIX, 2004), which has both a free open-source version as well as a more fully featured and FDA-approved version (which includes a certified PACS-viewer) which is appropriate if the tool is to be used for diagnostic purposes (Rosset et al., 2004). ClearCanvas Workstation (ClearCanvas Ontario, CA http://clearcanvas.ca) provides similar functionality for the Windows environment and also features both a paid FDA-approved version as well as a free open-source version which has been demonstrated to facilitate inter-rater agreement (Hsieh et al., 2013). ClearCanvas also supports plugins, making it adaptive to specific user needs. For example, our lab (Gutman et al., 2013) previously used a plugin that permits the use of the AIM markup language (Channin et al., 2009) for structured annotations. In addition to standard editions, ClearCanvas also offered a beta-release version that supported an integrated Web Viewer, although the current status of that project is unclear.

Another useful open-source project is the Medical Imaging Interaction Toolkit (MITK) which offers the user data management, advanced visualization, and interactive functions (http://www.mitk.org/MITK). The basic framework offers advantages of both Insight Toolkit (ITK) and Visualization Toolkit (VTK) and supports a wide variety of application plugins (some open-sourced, others not) to customize the user experience. For example, the “Iso Surface” plugin interpolates user-defined pixel selections and creates surface structures on regions of interest and the “IGT Tracking” plugin allows one to connect a tracking device to the image and record the resulting tracking location data. One aspect worth highlighting is the MITK Diffusion Imaging component which offers a suite of visualizations including fiber tractography, Q-Ball reconstruction, and Fiberfox to generate complex white matter tissue models.

InVesalius, a Brazilian program now in its third version, is another convenient tool used to view DICOM files from both CT and MRI protocols. It offers wide versatility and can be run on MS Windows, GNU Linux, and soon MacOS X systems. One of its main features is its detailed image reconstruction capability (http://svn.softwarepublico.gov.br/trac/invesalius/wiki/InVesalius/Screenshots).

Other lighter weight image viewers exist as well, such as the NIH ImageJ program (http://rsb.info.nih.gov/ij/), which

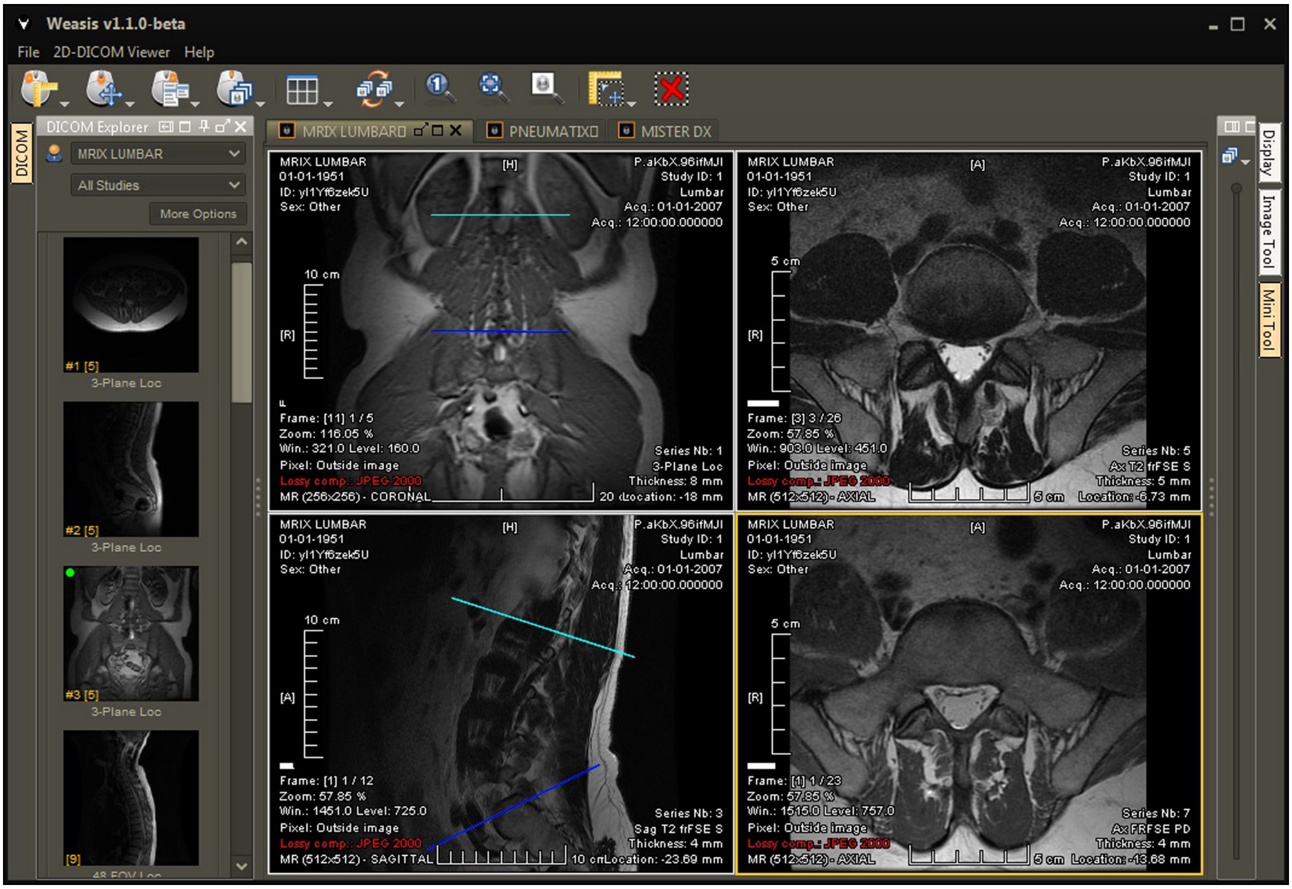

Weasis v1.1.0-beta
File 2D-DICOM Viewer Help
DICOM Explorer
MRIX LUMBAR
All Studies
More Options
#1 [5] 3-Plane Loc
#2 [5] 3-Plane Loc
#3 [5] 3-Plane Loc
[9] 48 FOV Loc
MRIX LUMBARD
PNEUMATIXD
MISTER DX
DICOM
Display
Image Tool
Mini Tool
MRIX LUMBAR 01-01-1951 ID: yl1Yf6zek5U Sex: Other
P.aKbX.96ifMJI Study ID: 1 Lumbar Acq.: 01-01-2007 Acq.: 12:00:00.000000
[H]
[A]
[R]
10 cm
5 cm
Frame: [11] 1 / 5 Zoom: 116.05 % Win.: 321.0 Level: 160.0 Pixel: Outside image Lossy comp.: JPEG 2000 MR (256x256) - CORONAL
Series Nb: 1 3-Plane Loc Thickness: 8 mm 20 cmLocation: -18 mm
Frame: [3] 3 / 26 Zoom: 57.85 % Win.: 903.0 Level: 451.0 Pixel: Outside image Lossy comp.: JPEG 2000 MR (512x512) - AXIAL
Series Nb: 5 Ax T2 frFSE S Thickness: 5 mm 5 cm Location: -6.73 mm
Frame: [1] 1 / 12 Zoom: 57.85 % Win.: 1451.0 Level: 725.0 Pixel: Outside image Lossy comp.: JPEG 2000 MR (512x512) - SAGITTAL
Series Nb: 3 Sag T2 frFSE S Thickness: 4 mm 10 cmLocation: -23.69 mm
Frame: [1] 1 / 23 Zoom: 57.85 % Win.: 1515.0 Level: 757.0 Pixel: Outside image Lossy comp.: JPEG 2000 MR (512x512) - AXIAL
Series Nb: 7 Ax FRFSE PD Thickness: 4 mm 5 cm Location: -13.68 mm

FIGURE 1 | User interface of Weasis, a program available through the DCM4CHE imaging collection. Weasis offers several tools to facilitate viewing large amounts of clinical data, such as line drawing, measurements, and a magnifying window. For more screenshots, visit http://www.dcm4che.org/confluence/display/WEA/Home.

supports DICOM (among many other formats), and IrfanView (http://www.irfanview.com/).

COINS, MIDAS, PESSCARA, XNAT: RESEARCH INFORMATICS PLATFORMS

Public imaging informatics systems are designed to complement the limitations of a clinically-focused PACS and allow for the smooth exchange of data between investigators to facilitate research. They also generally support other image formats besides DICOM that are commonly used as intermediates during image analysis.

COINS—Collaborative Informatics and Neuroimaging Suite

The COllaborative Informatics and Neuroimaging Suite (COINS) was developed at the Mind Research Network headquartered in Albuquerque, New Mexico and currently holds imaging data from more than 20,000 participants (Scott et al., 2011). COINS is an online portal where imaging data, as well as reports, annotations, and billing data, can be automatically archived into the system via a DICOM receiver. Additional accompanying data from interviews, questionnaires, and neuropsychological tests can also be entered through a web application called Assessment Manager (ASMT, **Figure 2**). COINS also features a Data Exchange Tool designed to facilitate communication by allowing de-identified neuroimaging datasets with associated metadata to be shared between collaborating research groups. In addition, a Medical Imaging Computer Information System (MICIS) component allows smooth project creation and participant enrollment and management, making COINS a useful tool in human research and clinical studies. In addition to overcoming complicated challenges involved with human subjects and PHI, principal investigators are able to set permissions assigning different levels of access according to various guidelines set by the Institutional Review Board.

MIDAS—The Multimedia Digital Archiving System

The Midas Platform is a PHP-based data storage system designed to facilitate computational scientific research by integrating data from a variety of sources (http://www.midasplatform.org/) (**Figure 3**). MIDAS was developed and is maintained by the same developers behind the VTK toolkit, which is commonly used throughout many imaging analysis modalities. The open source software, now in its 3.2.8 version, indexes data sources from imaging databases and visualization tools. The Midas framework can then query the back-end database. One benefit of

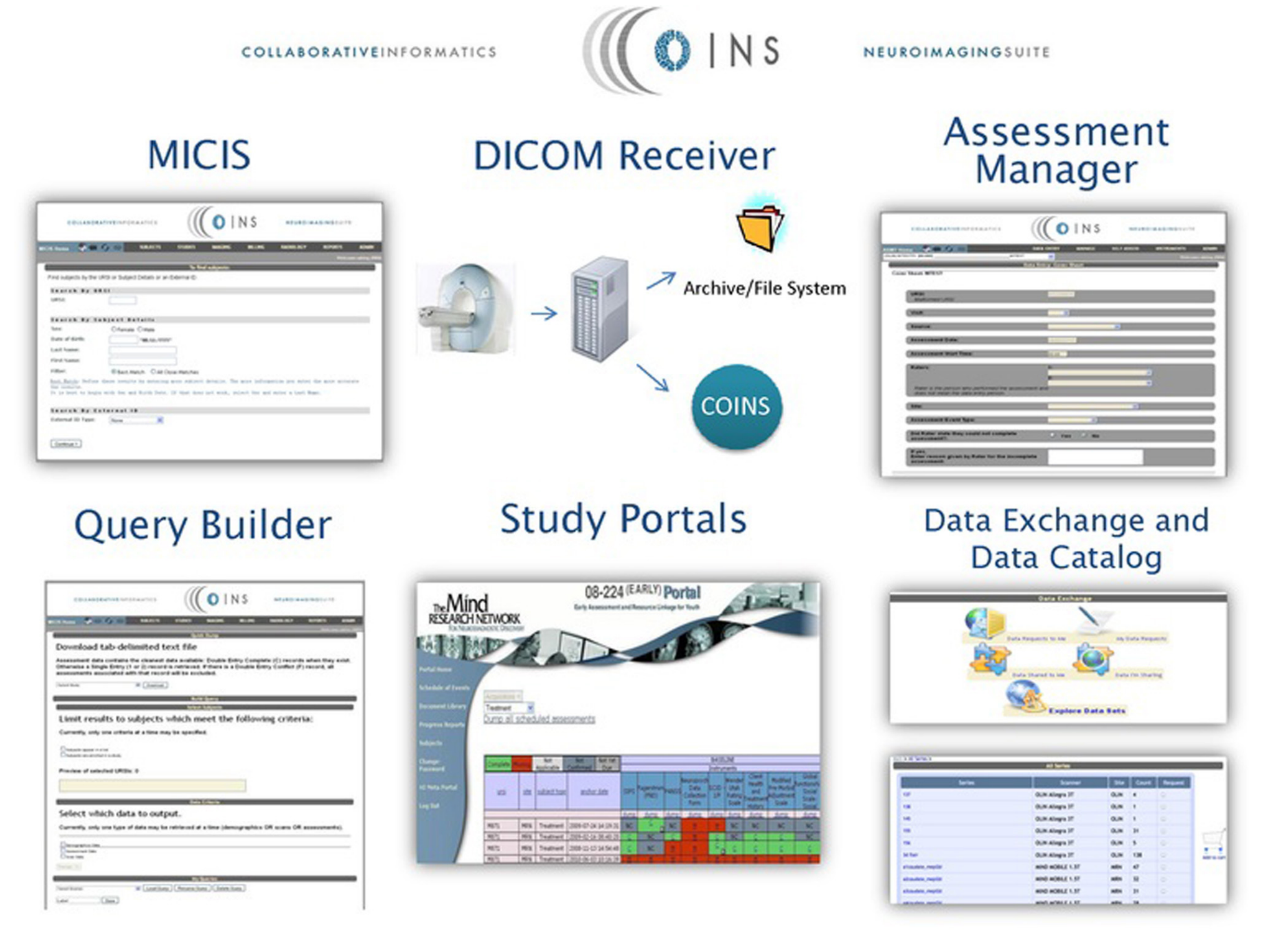


FIGURE 2 | An overview of the COINS web-based neuroimaging software suite. "COINS Tools: MICIS—Participant enrollment and management, MRI imaging data import, Scan annotation and behavioral data management, Radiology review event reports, Scan time billing. DICOM Receiver—Automates image archiving to file system and storage of meta-data to MICIS. Assessment Manager—Single and double entry as well as self-assessment. Query Builder—Secure, *ad-hoc* querying of single and cross-site studies for assessments, scans and demographics. Study Portals—Progress reports for subject tracking, shareable documents (study measures, meeting notes, etc.). Data Exchange with Data Catalog—Browse, request and share data, available for imaging data and clinical assessments, tracks data requests and keeps an inventory of data." (http://neuroinformatics2012.org/abstracts/coins-collaborative-informatics-neuroimaging-suite-give-get-collect).

the platform is its ability to be highly customized with various plugins to individually tailor the program to specific research needs.

PESSCARA—Platform to Enable Shared Scientific Computing And Research Advances

PESSCARA is a platform based on open-source resources and combines four important components for the conduct of science. The first components are image data and metadata stores, obviously critical starting points. We use the open source DCM4CHEE software to provide the mechanism for receiving and sending DICOM data. In most cases, there are processing steps applied to the medical images, including filtering, registration, segmentation, etc. These steps create new versions of the images, or add metadata about the images. Content management systems were built to do exactly these functions, and so we have leveraged the TACTIC CMS as the second major component of PESSCARA. It is open-source and has a Python application programming interface (API) to allow automation of many steps.

The third component of PESSCARA is an algorithm development environment. For that, we selected iPython Notebooks. Python has become the major programming language of science because of powerful libraries that can efficiently handle most tasks, because the language itself is easy to understand and has free interpreters for all major operating systems, and because the iPython Notebook provides a flexible way to develop, share, and document algorithms. Python has powerful image processing libraries, and also powerful data analysis tools. A mechanism for documenting the complete processing flow, including input data, processing steps, and results is key to shareable science. The fourth and final component is a results repository that allows a

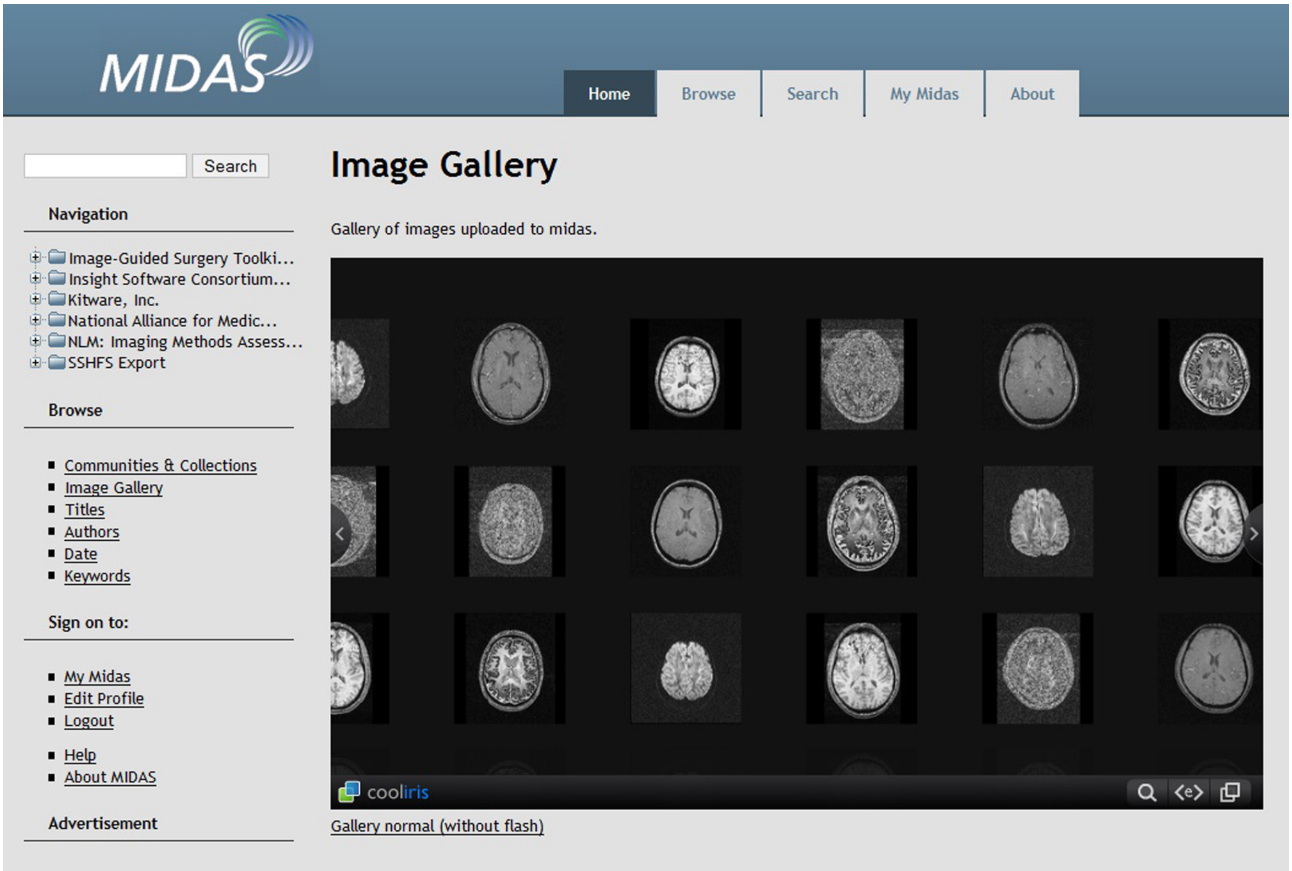


Image Gallery

Gallery of images uploaded to midas.

FIGURE 3 | Screenshot of the Midas Platform. Midas is a web-based toolkit that allows facilitated review of clinical imaging data through digital storage, online reporting, interactive visualization, and server-slide processing (http://www.midasplatform.org/MIDAS/resources/toolbox.html).

user to document and share all of the parts. This can allow other investigators to validate the results, as well as to test the same processing steps on other data sets or other processing algorithms on the same data set. We are currently in the process of posting the code and a virtual machine instance of this framework (http://PESSCARA.org).

XNAT—The eXtensible Neuroimaging Archive Toolkit

XNAT is an open source imaging informatics platform developed at Washington University in St. Louis and was designed for the storage and management of large heterogeneous imaging data sets to facilitate neuroradiological research (http://xnat.org/) (Marcus et al., 2007). The extendibility allows each research group to customize an "instance" and extend the basic application to suit their needs. Originally developed to store Phillips PAR files, the application now has a robust DICOM image management system and also allows storage of other common imaging formats (NII, Analyze, MGZ, etc). XNAT provides key functionality such as uploading and downloading data in various formats, organizing and sharing data, and customizing security and access to the data. In XNAT, users are able to save the original or modified files to disks or send them across a network to a DICOM C-STORE service class provider, such as a PACS or another XNAT instance. In addition, XNAT provides a means to view the data using a built-in Java-based DICOM viewer. The viewer relies on plugins to implement image-type-specific functionality and additional plugins can be developed and integrated to customize the viewer.

One of the key advantages of XNAT and similar systems relative to a more traditional PACS-based image management system is the flexibility provided in "tagging" data. Certain features that are critical in a research setting, such as the ability to associate certain patients with certain research protocols, are not easily handled in a typical PACS. A PACS viewer is usually organized around selecting by patient name, doctor who ordered the study, imaging modality or scan date—data that is oftentimes superfluous outside of the clinic. Once image sets are tagged in XNAT, patients can be neatly organized into projects and sorted by name, ID, or other relevant features.

WEB-BASED VISUALIZATIONS

The X Toolkit

Emerging technologies, including webGL and increased performance of JavaScript engines, now allow both 2D and 3D image manipulation on the client side. The X Toolkit (XTK, http://www.goXTK.com) and BrainBrowser (https://brainbrowser.cbrain.mcgill.ca/) are two popular tools that allow visualization and interaction with both 2-D (i.e., texture files .png, .jpg) and 3-D volumes, as well as the support of masks, tractography results, and/or label maps.

The X Toolkit is available on GitHub and can be used to visualize a wide spectrum of physiological phenomena ranging from white matter cortical connections, aneurysm characteristics, and knee morphologies. Of note, an interesting JavaScript library, jsdicom, also supports native DCM reading of DICOM files and is available on GitHub (https://github.com/Infogosoft/jsdicom).

An attractive feature of the XTK platform is that apart from being a native JavaScript library that directly parses DICOM files directly (as opposed to requiring a server side plugin to transcode .dcm files into .jpg/.png images), XTK can also support a number of other common neuroimaging formats such as several compressed and uncompressed formats of DICOM files (.nrrd, .nii, .nii.gz, .mgz, .dcm, etc.) as well as files from higher level MR processing (.trk, .stl, .fsm., .label, etc.) commonly used in image analysis research.

Several implementations of The X Toolkit include the AneuRisk Web repository (http://mox.polimi.it/it/progetti/aneurisk/) (Ford et al., 2009) and SliceDrop.org (**Figure 4**). The LONI group (formerly of UCLA, now of UCSC) has developed an extension of SliceDrop that further supports drawing ROIs directly within the XTK framework (http://users.loni.ucla.edu/∼pipeline/viewer/). A pediatric brain atlas, also built using the XTK visualization platform, further demonstrates the power of this framework (http://fnndsc.github.io/babybrain/, **Figure 5**). Another notable web based viewer is Papaya (http://github.com/rii-mango/Papaya), based on a similarly functioned Java client (http://en.wikipedia.org/wiki/Mango_(software)).

MATERIALS AND METHODS

XNATView was designed as a light-weight version of the bundled XNAT image viewer, which is in essence ImageJ (http://rsb.info.nih.gov/ij/index.html). As mentioned above, this current bundled application is Java-based, and we had difficulty on some of our machines installing the proper version of Java and consistently loading the application.

The initial prototype of XNATView was developed using the Adobe Flex framework, but the current implementation is now written in native JavaScript. XNATView uses a Representational State Transfer (REST) (Fielding, 2000) API to query the XNAT database below, capitalizing on the PyXNAT (Schwartz et al., 2012) library. The back-end functionality is written in Python and the user interface is primarily written in jQueryUI (http://jqueryui.com/).

LEVERAGING XNAT's REST INTERFACE TO DEVELOP A CUSTOMIZABLE IMAGE MANAGEMENT SYSTEM

One of the most powerful aspects of the XNAT framework is the introduction of a REST-based API, which allows programmatic access to the available imaging data. In developing XNATView, we have exploited this capability to develop our own image viewer and web-based GUI for image navigation. The PESSCARA framework also supports REST-based queries, allowing us to leverage the XNATView architecture and generalize it to produce a more flexible zero footprint image viewer.

While the rich metadata which XNAT provides related to scan times, quality, echo time, etc. is important to be able to access by "power users," the REST interface allows our lightweight viewer to sit on top and to abstract many details which may be overwhelming for the average user. In this way, XNATView trades some of the functionality for performance by allowing its users to be able to quickly view the imaging data and accompanying metadata while providing basic image processing tools such as contrast and zoom. The average researcher is allowed to tailor his or her interface and expose select data elements to their user base to allow for cleaner image viewing and annotation. The images are also cached, which clears some of the load away from the XNAT back-end.

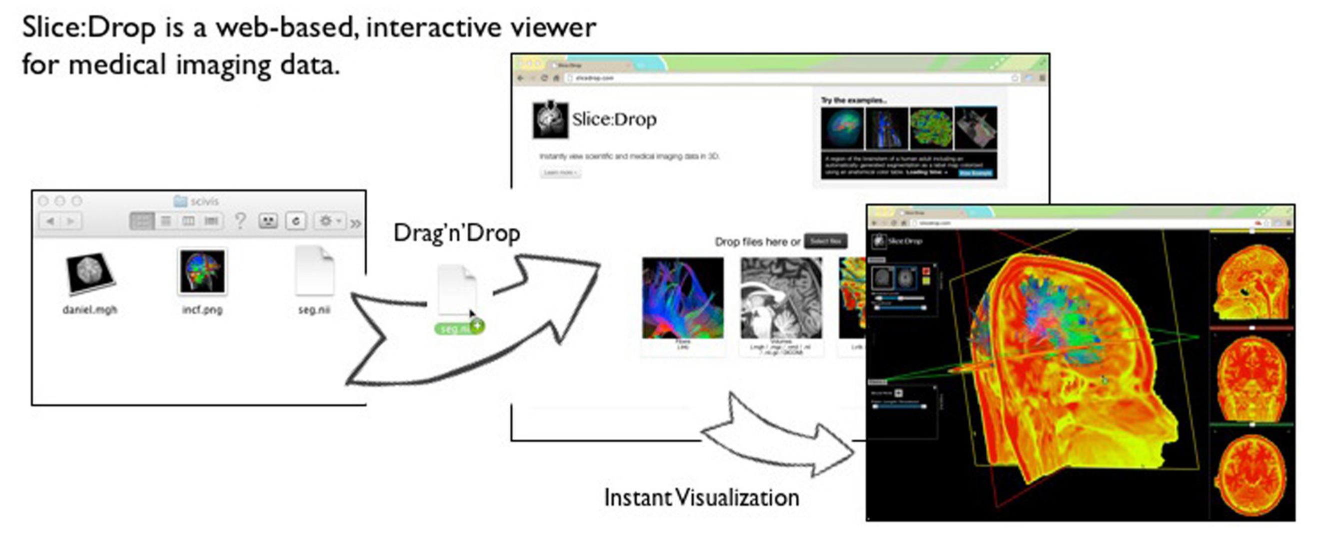


FIGURE 4 | Slice:Drop An interactive visualization tool that allows users to instantly visualize imaging data from a wide variety of compatible imaging formats (available at http://slicedrop.com/).

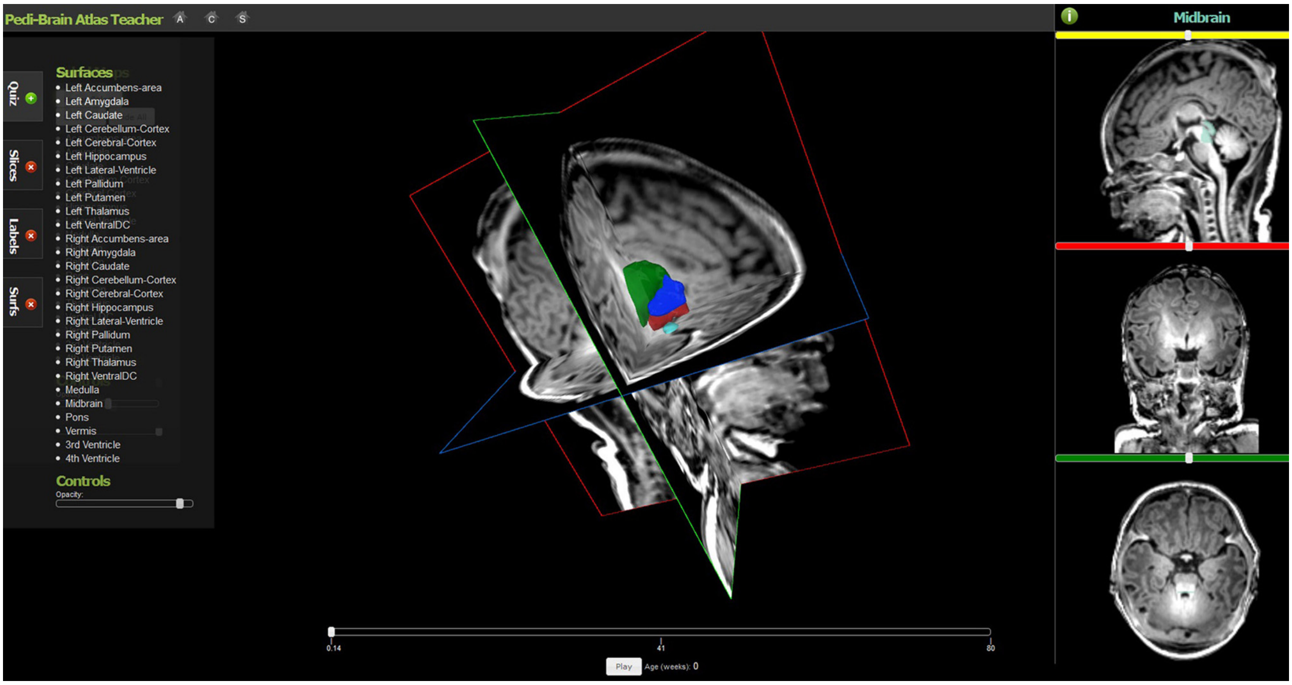


FIGURE 5 | The Pedi-Brain Atlas Teacher, an interactive visualization tool for pediatric brain tumors on MRI (available at: http://fnndsc.github.io/babybrain/).

PyXNAT

XNATView’s back-end functionality is written in Python and communicates with XNAT via PyXNAT (Schwartz et al., 2012). PyXNAT is a Python library that wraps around the RESTful Web Services provided by XNAT and aims to bridge the communication with an XNAT server and provides an object-oriented approach to querying the data. PyXNAT, combined with other scientific libraries available in Python, allows the user to query, change metadata, upload, and download files in a structured and intuitive way. For example, if we want to get a list of projects in a given XNAT instance (identified by an instance string, username, and password), we would type the following command:

```
xnat = Interface(server=instance,
  user=username, password=password,
cachedir = os.path.join(os.path.expanduser
  (’~’),’XNATVIEW/.store’))
project_list = xnat.select.projects().get()
```

The variable `project_list` will now contain all the names of projects in the specific XNAT instance. PyXNAT preserves the hierarchy and data organization in its API, so if we would want to get a list of subjects in a particular project we would type:

```
xnat.select.project(project_name).
  subjects().get(’label’)
```

Downloading and reading DICOM files is also simplified with PyXNAT. We use the `dicom` package available in Python to read DICOM tags and PyXNAT API to download the files from the online archive to our systems. The following code downloads the DICOM files associated with a scan and reads the instance number in each slice.

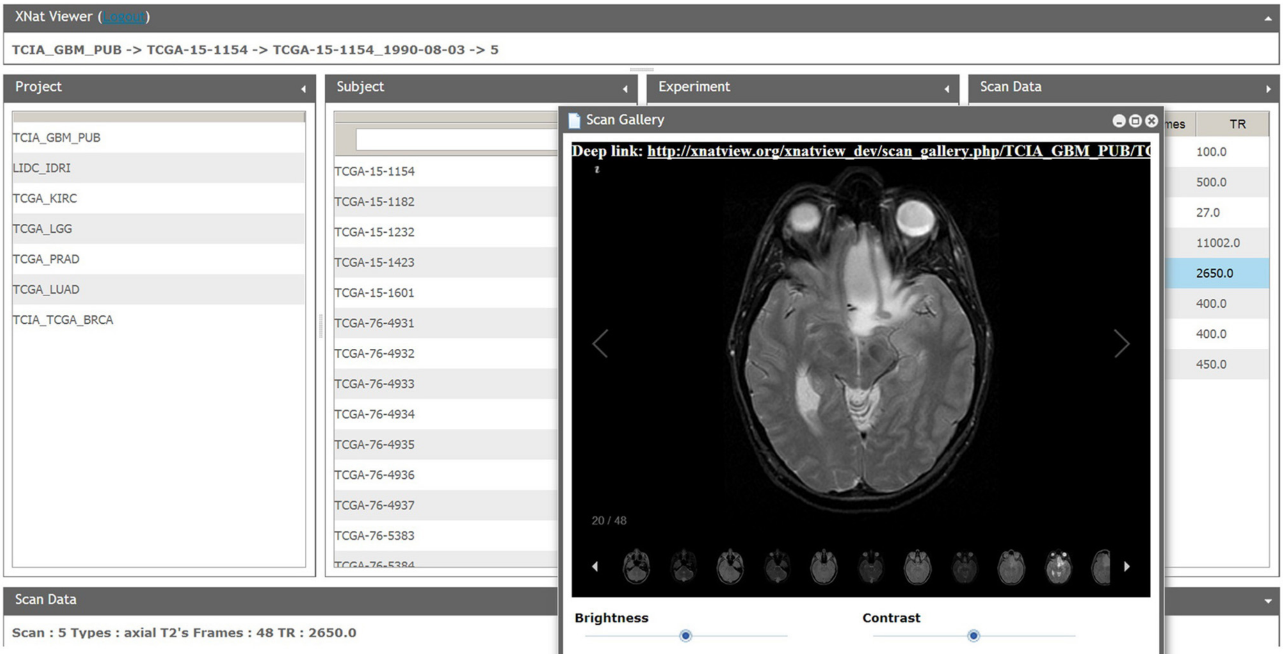
```
for each_file in scan.resource(’DICOM’).
  files():
    #download from XNAT in tempDir
    path = os.path.join(tempDir,each_file.
      id())
    each_file.get(path,False)
    #read DICOM tags
    dicomData = dicom.read_file(path)
    tag = str(dicomData.InstanceNumber)
```

`Scan` is an object that contains information specific to a particular scan, `tempDir` is a path on the local system, and `dicom` is the Python package (`import dicom`).

RESULTS

XNATView ORGANIZATION

One of the most important characteristics of XNATView is its very simple and intuitive user interface that allows a user to browse among tens of thousands of images from one centralized location. The data is organized using the same hierarchical concept used in XNAT—the data is grouped left to right according to projects, subjects, experiments, and scan type (**Figure 6**). The home user interface also allows users to view basic metadata such as patient age and medication regimen associated with the scan as well as easily view, compare, and analyze multiple scans at once.



TCIA_GBM_PUB -> TCGA-15-1154 -> TCGA-15-1154_1990-08-03 -> 5

TCIA_GBM_PUB
LIDC_IDRI
TCGA_KIRC
TCGA_LGG
TCGA_PRAD
TCGA_LUAD
TCIA_TCGA_BRCA

TCGA-15-1154
TCGA-15-1182
TCGA-15-1232
TCGA-15-1423
TCGA-15-1601
TCGA-76-4931
TCGA-76-4932
TCGA-76-4933
TCGA-76-4934
TCGA-76-4935
TCGA-76-4936
TCGA-76-4937
TCGA-76-5383

Deep link: http://xnatview.org/xnatview_dev/scan_gallery.php/TCIA_GBM_PUB/T

20 / 48

100.0
500.0
27.0
11002.0
2650.0
400.0
400.0
450.0

Scan : 5 Types : axial T2's Frames : 48 TR : 2650.0

FIGURE 6 | XNATView user interface. Users can select a specific scan narrowing search criteria from project, subject, session, and scan from left to right. Selected scans can be custom tiled to suit specific viewing needs, scrolled between slices, and viewed using various image settings such as brightness and contrast.

DATA FEDERATION

One interesting aspect of XNATView is the potential to allow multiple systems (XNAT based or other) to present a single federated view of available data sources. Specifically, the ability to exploit the powerful REST interface accessible via PyXNAT allows individual labs to maintain their own imaging repository but share higher level attributes (available subjects, projects, etc.) while still maintaining access to the underlying images.

By facilitating the need to send and manage multiple files for review by collaborators, these interactive tools allow immediate feedback from collaborators. Thus, in addition to improving upon traditional image communication methods involving static images or single slice views, we currently support the communication of images adjusted for brightness/contrast, opacity, color, and other properties of image overlays/masks.

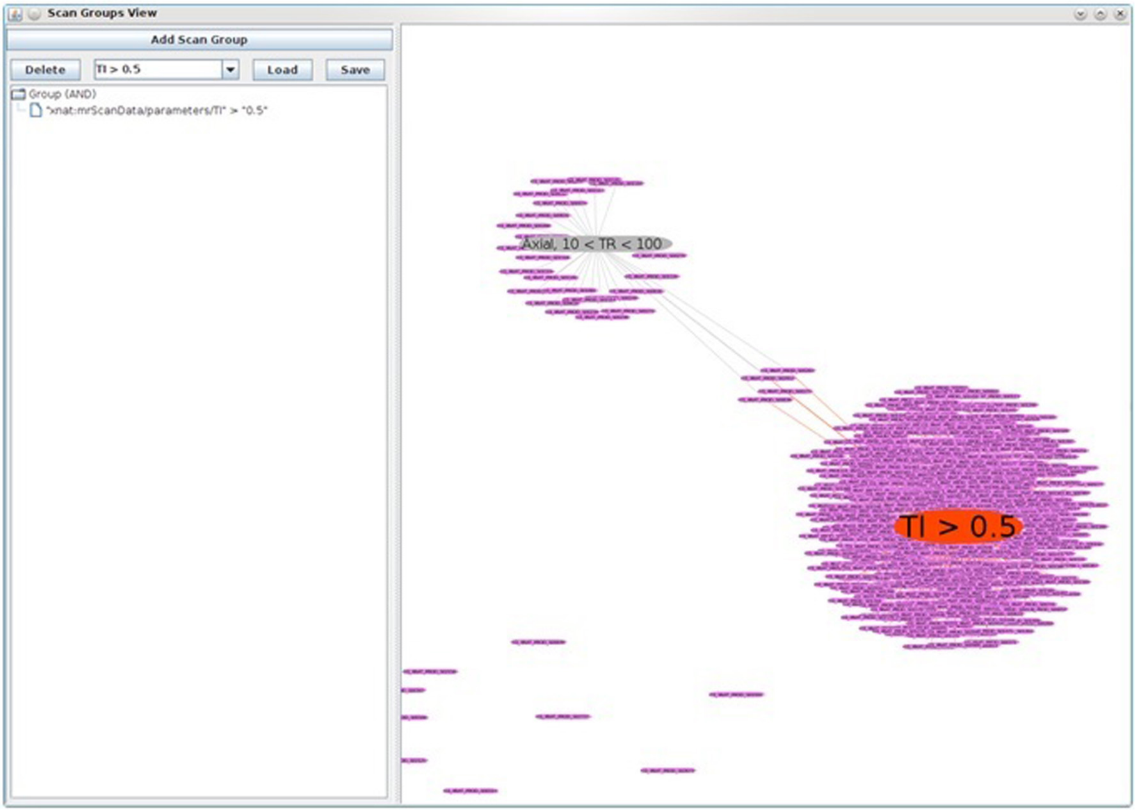
ENHANCED DATA ACCESS

Another valuable feature we have incorporated is the ability to provide a “deep link” to a file. When a user loads an image, a URL appears at the top of the screen; the user can email a colleague the URL and be directly sent to an image of interest. This could be further extended to maintain various desired settings (e.g., contrast, brightness, zoom). As we further integrate the ability to visualize image masks, this feature will become even more useful.

As another example of this flexibility, we have developed a data-finding utility we internally dubbed “XNAT Soup.” As we were trying to group and visualize image series with similar scan parameters, we developed an application that allows us to visualize collection of images with similar scan properties. XNAT Soup includes some standard visualizations, such as scatterplots, but the main feature is a novel visualization for identifying relationships between groups of scans. The visualization uses a force-based layout that supports dragging, panning and zooming. Each subject is represented as a single node. A repulsive force causes subject nodes to spread out so as not to appear on top of one another and a drag force prevents this from causing them to fly out of view.

XNAT Soup utilizes XNAT’s REST-based search-engine API to allow the user to execute any scan queries that are supported by XNAT itself. Each search query is added to the visualization as a new node, which we call a *scan group node*. The size of each scan group node is determined by the number of scans included in the group. For each subject node that has scans in the scan group, a spring-force edge is added between the subject node and scan group node. The strength of the spring force is determined by the percent of scans in the scan group that belong to that subject.

The end result is that subjects related to a scan group are drawn close to it and follow it when the scan group node is moved around. Any node can be double-clicked to force its position to remain stationary. When there are multiple scan groups present, some subjects will appear between the two when they have scans in both groups. This can be used to identify groups of subjects with particular characteristics. Hovering over any node reveals additional details in a tooltip. **Figure 7** shows an example where two scan groups are shown: (1) the smaller group, in gray, includes only Axial scans with a TR value between 10 and 100, while (2) the larger group, in red, includes all scans with a TI value above 0.5. From this view, we can see that there are four subjects with scans in both groups.



TI > 0.5

Group (AND)
"xnat:mrScanData/parameters/TI" > "0.5"

FIGURE 7 | XNAT Soup, a data finding utility used to visualize collections of images with similar scan properties. In this figure, two scan groups are shown: (1) the smaller group, in gray, includes only axial scans with a TR value between 10 and 100, while (2) the larger group, in red, includes all scans with a TI value above 0.5. Note also the four subjects in between, who have scans from both groups.

ZERO-INSTALLATION FEATURE

Finally, to keep up with rapidly evolving technology, we are quickly moving toward a zero-installation model for MRI visualization and manipulation. This allows for a centralization of image data which helps with the problem of file versioning as the original file is not downloaded but only streamed from the server to the webclient. Although there are pragmatic concerns, such as the ubiquity of outdated Internet Explorer installations on many machines, the technical burden of sharing and interacting with images is rapidly decreasing. This then offers the possibility of interaction and feedback from colleagues with a wide degree of technical expertise and further fosters collaboration and knowledge discovery.

XNATView OPERATION

XNATView allows the user to choose an XNAT instance and log-in using XNAT credentials for that instance. The flexibility of the REST-based services in XNAT allows the XNATView to run without any modification of XNAT itself, and can communicate with any accessible XNAT back-end. In addition, various visualization plugins and our basic 2-D slice viewer as well as experimental support for an XTK based 3D visualization tool are supported by our XNATView. The basic XNATView interface supports Internet Explorer 8 with limited functionality, which is often the standard browser on many hospital and clinic settings. A public instance of XNATView is available at http://xnatview.org/ (This version offers basic user options and is available by clicking "guest" at http://xnatview.org/), which mirrors publically available data provided by the Cancer Imaging Archive (Prior et al., 2013).

DISCUSSION

To support various neuroimaging research demands in our lab, we have developed XNATView, a tool that interfaces with XNAT and leverages the REST layer which XNAT exposes for programmatic data access. As we further develop this, we hope to generalize this software into a Zero Foot Print Image Viewer (ZFIV) which can support multiple back-ends (e.g., not just XNAT or PESSCARA).

XNATView capitalizes on the functionalities of PyXNAT and serves as a lightweight interface that allows easy visualization of a wide range of image series along with metadata. As a result, users are able to review thousands of images from one centralized location, which has the potential to improve data sharing and collaboration (Walden et al., 2011). Some advantages of our implementation are the ability to provide "deep links" allowing users direct access to a particular scan/session, the potential to provide federated views to multiple backends (XNAT or other), a simple UI, Internet Explorer support back to version 8 (with somewhat reduced functionality), and removing the dependency on Java.

It is important to highlight the extreme flexibility that both PyXNAT and XNAT allow via the REST based user interface. REST is a powerful tool used to access, query, retrieve and convert database entries and we chose this platform based on its ease of utilization and functional flexibility. In fact, it has been used in several applications ranging from displaying bioinformatics data from sequence alignment data (Katayama et al., 2010) to assisting physicians in drug prescription decisions (Bianchi et al., 2013).

Therefore, while we feel the application is itself of interest, perhaps the most important aspect of this work is the ability to leverage the power of REST-based mechanisms to allow "mash-ups." In essence, XNATView is a simple thumbnail gallery, similar to the multitude of image viewers available. The ability to expose data via REST, however, allows the end-user to repurpose and abstract many of the functions of the underlying tool (XNAT) to suit their own needs. As discussed above, our current work with PESSCARA, which also supports REST based image query and retrieval, can be similarly attached. Of note, the name XNATView reflects the initial implementation of this framework, although as we enable other back-ends, ZFIV may be a more appropriate moniker.

CODE AVAILABILITY

The code and initial application is available at our github site [https://github.com/dgutman/ZeroFootPrintImageViewer_XnatView].

FUNDING INFORMATION

This work was supported in part by PHS Grant UL 1RR025008 from the Clinical and Translational Science Award Program, National Institutes of Health, Contract No HHSN261200800001E from the national Cancer Institute, National Institutes of Health, and the Georgia Research Alliance.

ACKNOWLEDGMENTS

We would like to acknowledge Dhananjaya Somanna and Lee A. D. Cooper PhD for help developing the initiation prototype of this project.

REFERENCES

Bianchi, L., Paganelli, F., Pettenati, M. C., Turchi, S., Ciofi, L., Iadanza, E. et al. (2013). Design of a RESTful Web information system for drug prescription and administration. *IEEE J. Biomed. Health Inform.* 18, 885–895. doi: 10.1109/JBHI.2013.2282827

Bryan, S., Weatherburn, G. C., Watkins, J. R. and Buxton, M. J. (1999). The benefits of hospital-wide picture archiving and communication systems: a survey of clinical users of radiology services. *Br. J. Radiol.* 72, 469–478.

Cancer Genome Atlas Research, N., Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R., Ozenberger, B. A., et al. (2013). The cancer genome atlas pan-cancer analysis project. *Nat. Genet.* 45, 1113–1120. doi: 10.1038/ng.2764

Channin, D. S., Mongkolwat, P., Kleper, V. and Rubin, D. L. (2009). The annotation and image mark-up project. *Radiology* 253, 590–592. doi: 10.1148/radiol.2533090135

DICOM@OFFIS., (2013). *DCMTK - DICOM Toolkit*. Available online at: http://dicom.offis.de/dcmtk.php.en (Accessed January 4, 2014).

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-Based Software Architectures*. Doctor of Philosophy, University of California, Irvine, CA.

Ford, M. D., Hoi, Y., Piccinelli, M., Antiga, L. and Steinman, D. A. (2009). An objective approach to digital removal of saccular aneurysms: technique and applications. *Br. J. Radiol.* 82, S55–S61. doi: 10.1259/bjr/67593727

Gutman, D. A., Cooper, L. A., Hwang, S. N. Holder, C. A., Gao, J., Aurora, T. D., et al. (2013). MR imaging predictors of molecular profile and survival: multi-institutional study of the TCGA glioblastoma data set. *Radiology* 267, 560–569. doi: 10.1148/radiol.13120118

Hsieh, J., Honda, A. F., Suarez-Farinas, M., Samson, C. M., Kedhar, S., Mauro, J., et al. (2013). Fundus image diagnostic agreement in uveitis utilizing free and open source software. *Can. J. Ophthalmol.* 48, 227–234. doi: 10.1016/j.jcjo.2013.02.010

Katayama, T., Nakao, M. and Takagi, T. (2010). TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics Web services. *Nucleic Acids Res.* 38, W706–W711. doi: 10.1093/nar/gkq386

Marcus, D. S., Olsen, T. R., Ramaratnam, M. and Buckner, R. L. (2007). The extensible neuroimaging archive toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34. doi: 10.1385/NI:5:1:11

OSIRIX. (2004). *OsiriX Imaging Software*. Available online at: http://www.osirix-viewer.com/ (Accessed January 4, 2014).

Prior, F. W., Clark, K., Commean, P., Freymann, J., Jaffe, C., Kirby, J., et al. (2013). TCIA: an information resource to enable open science. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2013, 1282–1285. doi: 10.1109/EMBC.2013.6609742

Rosset, A., Spadola, L., and Ratib, O. (2004). OsiriX: an open-source software for navigating in multidimensional DICOM images. *J. Digit. Imaging* 17, 205–216. doi: 10.1007/s10278-004-1014-6

Schwartz, Y., Barbot, A., Thyreau, B., Frouin, V., Varoquaux, G., Siram, A., et al. (2012). PyXNAT: XNAT in python. *Front. Neuroinform.* 6:12. doi: 10.3389/fninf.2012.00012

Scott, A., Courtney, W., Wood, D., de la Garza, R., Lane, S., King, M., et al. (2011). COINS: an innovative informatics and neuroimaging tool suite built for large heterogeneous datasets. *Front. Neuroinform.* 5:33. doi: 10.3389/fninf.2011.00033

Walden, A., Nahm, M., Barnett, M. E., Conde, J. G., Dent, A., Fadiel, A., et al. (2011). Economic analysis of centralized vs. decentralized electronic data capture in multi-center clinical studies. *Stud. Health Technol. Inform.* 164, 82–88. doi: 10.3233/978-1-60750-709-3-82

Warnock, M. J., Toland, C., Evans, D., Wallace, B., and Nagy, P. (2007). Benefits of using the DCM4CHE DICOM archive. *J. Digit Imaging* 20(Suppl. 1), 125–129. doi: 10.1007/s10278-007-9064-1

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 27 January 2014; accepted: 29 April 2014; published online: 27 May 2014.
Citation: Gutman DA, Dunn WD Jr, Cobb J, Stoner RM, Kalpathy-Cramer J and Erickson B (2014) Web based tools for visualizing imaging data and development of XNATView, a zero footprint image viewer. Front. Neuroinform. ***8****:53. doi: 10.3389/fninf.2014.00053*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 05 May 2014
doi: 10.3389/fninf.2014.00042

Interactive 3D visualization of structural changes in the brain of a person with corticobasal syndrome

Claudia Hänel[1]*, Peter Pieperhoff[2], Bernd Hentschel[3], Katrin Amunts[2,3] and Torsten Kuhlen[3]

[1] JARA - High Performance Computing, IT Center - Computational Science and Engineering, Computer Science Department, Virtual Reality Group, RWTH Aachen University, Aachen, Germany
[2] JARA - Translational Brain Medicine, Institute of Neuroscience and Medicine (INM-1), Research Centre Jülich, Jülich, Germany
[3] C. and O. Vogt Institute for Brain Research, Heinrich Heine University Düsseldorf, Düsseldorf, Germany

Edited by:
John Van Horn, University of California at Los Angeles, USA

Reviewed by:
Markus Butz, Forschungszentrum Jülich, Germany
Timothy Rittman, University of Cambridge, UK

****Correspondence:***
Claudia Hänel, JARA - High Performance Computing, IT Center - Computational Science and Engineering, Computer Science Department, Virtual Reality Group, RWTH Aachen University, Seffenter Weg 23, Aachen 52074, Germany
e-mail: haenel@vr.rwth-aachen.de

The visualization of the progression of brain tissue loss in neurodegenerative diseases like corticobasal syndrome (CBS) can provide not only information about the localization and distribution of the volume loss, but also helps to understand the course and the causes of this neurodegenerative disorder. The visualization of such medical imaging data is often based on 2D sections, because they show both internal and external structures in one image. Spatial information, however, is lost. 3D visualization of imaging data is capable to solve this problem, but it faces the difficulty that more internally located structures may be occluded by structures near the surface. Here, we present an application with two designs for the 3D visualization of the human brain to address these challenges. In the first design, brain anatomy is displayed semi-transparently; it is supplemented by an anatomical section and cortical areas for spatial orientation, and the volumetric data of volume loss. The second design is guided by the principle of importance-driven volume rendering: A direct line-of-sight to the relevant structures in the deeper parts of the brain is provided by cutting out a frustum-like piece of brain tissue. The application was developed to run in both, standard desktop environments and in immersive virtual reality environments with stereoscopic viewing for improving the depth perception. We conclude, that the presented application facilitates the perception of the extent of brain degeneration with respect to its localization and affected regions.

Keywords: volume rendering, view-dependent visualization, virtual reality, deformation-based morphometry, neurodegeneration, atrophy

1. INTRODUCTION

The simultaneous 3D visualization of both, the outer surface and internal structures of the human brain in an intuitively graspable manner is still challenging. The pattern of gyri and sulci of the outer brain surface provides landmarks for at least coarse localization. Besides, internal brain structures must be distinguishable due to their high functional specificity. Moreover, in neuroscience it is desired to combine such representations with different kinds of additional field data; thresholded maps of such field data shall be integrated with structural data. In the present study, the following types of field data were superimposed on a magnetic resonance imaging (MRI) scan of a brain of a patient who suffers from the corticobasal syndrome (CBS): Structural MRI data, time dependent field data that quantify structural changes on the voxel level, and probabilistic maps of anatomical regions (cf. Zilles et al., 2002; Amunts et al., 2007). The structural change data were calculated by analyzing series of longitudinally acquired MRI data using deformation based morphometry (DBM, cf. Pieperhoff et al., 2008). New insights into brain regions that are affected by certain neurodegenerative diseases are enabled by exploration of occurring structural changes and its temporal progression.

Visualizations of these brain data by means of 2D sections are widely used. However, each of these sections provides only a small cutout of the brain. Thus, it is left to the observer to mentally merge the information into a 3D representation. In particular, it is difficult to relate the information given in a 2D section to the cortical surface of an individual brain, e.g., to identify individual sulci. To this end, an additional 3D visualization can be provided separately or in combination with sections. Several software tools are available for the side-by-side 2D and 3D visualization of brain data. For example, Brainvoyager QX (http://www.brainvoyager.com/) is a commercial software specialized for functional MRI and diffusion tensor imaging (DTI), but the extension of this program for other data modalities is not straightforward because of special visualization needs. OpenWalnut (http://www.openwalnut.org/) and MITK (www.mitk.org) are open source toolkits that can be extended via plugins, or even within the code basis. Whereas OpenWalnut is specialized on DTI data, MITK is a tool for the processing of more general medical data. Both applications offer a 3D visualization complemented by 2D sections in different orientations.

The 3D visualization of the human brain, however, raises particular difficulties that are usually not considered by standard software. A transparent brain surface representation might become difficult to comprehend when the surface is overlapping too many times along the view direction. On the contrary, visualizing the brain as an opaque surface will occlude major parts of itself. For example, Thompson et al. (2007) and Zhou et al.

(2013) use opaque surface renderings of the whole brain or of certain segmented structures that are colored by additional field data, such as volume change data or statistical scores. But any information inside or outside the rendered surfaces is discarded. To enable the representation of subcortical brain structures like the border between cortex and white matter, subcortical nuclei, or ventricles, 3D visualizations are often combined with up to three clipping planes (cf. Weber et al., 2008; Olabi et al., 2012). These planes define a clipping cuboid, in which the part of the brain inside this cuboid is removed and the clipping planes itself represent 2D sections. For the combined visualization of data from different modalities, the clipping can be limited to individual data sets as shown in Born et al. (2009) and Rieder et al. (2013), where only the anatomy inside the clipping cuboid is removed and fiber bundles stay visible. Additionally, a transparent representation of the brain in the cutout improves spatial perception of non-clipped structures.

Using a cuboid for the clipping, however, may require clipping too large parts of the brain, for example, when structures in the central part of the brain should be depicted. Therefore, a more flexible clipping geometry would be beneficial. As an alternative to the clipping planes, Rick et al. (2011) present a flashlight metaphor that enables the user to define interactively a cutout within the opaque volume visualization of the brain anatomy. Orientation, diameter, and depth of the clipping cone can be adjusted by the user. Still, this method is of limited use for elongated structures when observing the whole structure at once, because the diameter of the cone becomes unnecessarily large for the non-elongated direction and again too large parts of the brain might be clipped. Otherwise, a cone with a small diameter could be moved manually through the volume showing only a small part of the volume of interest (VOI) at once.

More data-driven visualization concepts are presented, e.g., by Hauser et al. (2001), Krüger et al. (2006), Bruckner et al. (2006), and Viola et al. (2004). These designs have in common that they locally decrease the opacity of occluding structures to show internal ones. Hauser et al. (2001) suggest to render different structures of a data set as individual objects separately in a first step. These objects are merged in a second step in order to give each of them a customized appearance in the final visualization. To focus on a small part of the whole volume, Krüger et al. (2006) use focus and context techniques and apply different weights, transparency functions, or color properties for focus and context objects. Bruckner et al. (2006) motivate their concept by hand-drawn illustration techniques and influence the focus mainly by defining the distance to the eye point and a gradient magnitude. The user can influence the sharpness of transition between clipped and visible structures, and the depth of clipping. The drawback of the three previously described techniques is the missing depth information for the VOI. Viola et al. (2004) resolve this problem with a technique using a conical cutout, that is comparable to the flashlight metaphor of Rick et al. (2011). The cutout shows anatomical information on its faces, thus giving depth information. This importance-driven volume rendering (IDVR) approach has the advantage that the cutout can be assigned to a particular structure, so that it can be automatically adjusted to the VOI's size. Furthermore, the cutout follows the view direction of the user and therefore stays always perfectly aligned. But for neuroscientific applications this technique might be further improved by providing additional information at the same depth as the VOI creating a section-like view onto the data and a data-depended clipping object for the deformation data.

Based on the previous findings we introduce two designs for the visualization of brain data with time dependant structural changes. The user interface of our visualization system had to enable an intuitive interaction and to provide an overview of the whole data in combination with detailed view of spatial relations of anatomical structures. The first design provides detailed anatomical information by means of a transparent anatomy of a whole MRI brain data set, whereby a 2D section can be interactively defined within this volume (cf. **Figure 1**). The second design extends the approach of Viola et al. (2004) by using a frustum of a cone as clipping object (cf. **Figure 2**) to provide more context information about nearby structures on the clipping planes (cf. **Figure 3**).

The rest of the paper is structured as follows: In section 2 image data as well as details of our visualization designs and interaction strategies are described. In section 3, the benefits of our implementation are discussed and in section 4 conclusions are drawn and an outlook onto possible future improvements is given.

2. METHODS

Before we describe the two visualization designs mentioned above in detail, the underlying data modalities are clarified. Furthermore, we show how the user interface of our visualization system aims for an intuitive interaction and provide both, an overview of the whole data and a detailed view of spatial relations.

2.1. DATA AND IMAGE ANALYSIS

In this work a series of T1-weighted MR-images of a single person was used as an exemplar. The images were acquired by Südmeyer et al. (2012) in the context of a longitudinal study on aging and neurodegenerative diseases. The voxel-size of the MR-images was $1 \times 1 \times 1\,\text{mm}^3$. These images were acquired at five points in time within a total interval of 26 months. The initial MR-image was segmented by deleting the value of every voxel not belonging to brain tissue. Segmentation masks were automatically generated by a procedure which was implemented in the program SPM (http://www.fil.ion.ucl.ac.uk/spm/) and afterwards manually corrected. Maps of volume changes,

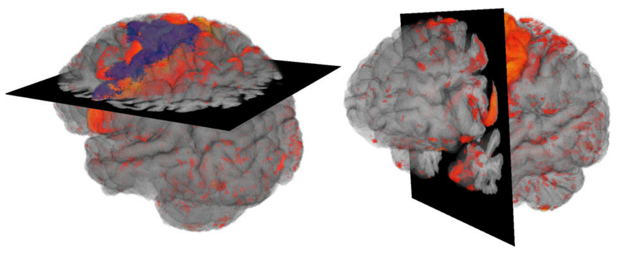


FIGURE 1 | Overview Design: Volume visualization showing brain degeneration (yellow/red) and the premotor cortex area (blue) in anatomical context (gray).

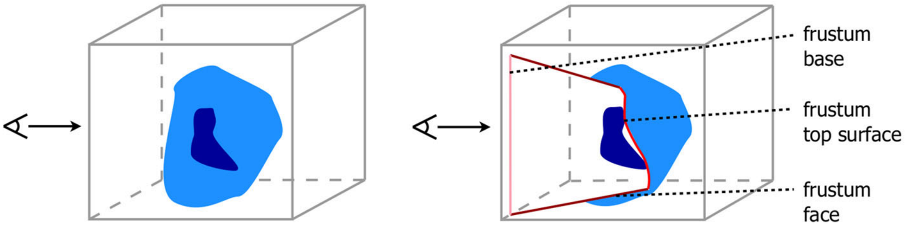


FIGURE 2 | View dependent frustum-like cutout into the volume (light blue) following the depth structure of the VOI (dark blue).

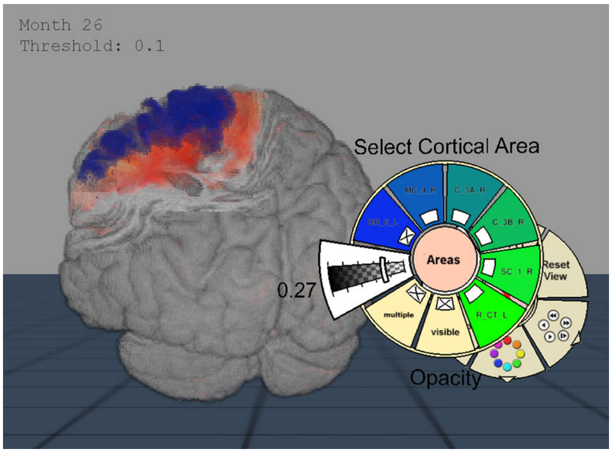


FIGURE 3 | Importance-Driven Volume Rendering Design: A view-dependent cutout is created to the premotor cortex area (blue). Furthermore, this screenshot shows the application when changing the opacity of the cortical area via pie menu.

which were superimposed to the structural image, were calculated by DBM in the following way. Each follow-up MR-image of the subject was non-linearly registered with the initial image by minimizing the voxel-wise squared intensity differences between both images, regularized by an elastic energy term which penalized non-biological distortions. The image registration yielded for each follow-up MRT image a deformation field that assigned to each voxel of the initial MRT image a vector that pointed to the corresponding position in the follow-up image. From this deformation field, a map of voxel-wise relative volume differences was derived. Further details of this analysis can be found in Pieperhoff et al. (2008). In order to visualize the temporal evolution of tissue degeneration fluently, volume change maps in-between the actual time points–month 0, 16, 20, 23, and 26–were interpolated to a total number of 27 data sets.

Maps of anatomical regions used here originate from the JuBrain Cytoarchitectonic Atlas (https://www.jubrain.fz-juelich.de). They were gained by cytoarchitectonic based parcellations in histological sections of post-mortem brains (cf. Zilles et al., 2002; Amunts et al., 2007).

The anatomical data, time dependent field data and cortical areas were used to develop the visualization designs presented below and were a use case to examine the supportive effect in the visual analysis of these data.

2.2. VISUALIZATION

We developed two different visualization designs to support the spatial understanding of the data. The first design used a transparent 3D representation of the anatomy and an opaque section. The second design was based on the IDVR algorithm as described in Viola et al. (2004) creating a view-dependent cutout around a defined VOI. In both designs, the degeneration of the brain tissue was visualized by means of time varying data, which were mapped to a red to yellow color map, with red meaning small and yellow large volume decline (cf. Südmeyer et al., 2012). Additionally, in order to identify the affected brain structures, maps of selected anatomical regions of the JuBrain atlas were included. For the visualization design described below, we had have to follow the requirement to present internal structural information with respect to external anatomy in a meaningful way. The use of volume rendering and an interactive adjustment of opacity values for each data set facilitated, for example, a visualization of structural changes caused by tissue atrophy and anatomical regions. Furthermore, the combined ray casting for all data in one volume renderer enabled a correct depth perception. Based on this, our first visualization design combined common modalities and was used particularly as an overview visualization, whereas the second design allowed for a detailed examination of a selective VOI.

2.2.1. Overview design

2D sections can be combined with 3D visualizations when using them as clipping planes (cf. Cabral et al., 1994; Rößler et al., 2006; Rick et al., 2011) to assist spatial orientation. Our proposed overview design was based on this idea: The brain anatomy was shown semi-transparently by the use of volume rendering and complemented by a 2D section of the original MRT data. Thus, both, the complex structure of the brain surface with gyri and sulci as well as internal regions remained visible, and no information in front of the section was lost as with clipping planes. To give an overview on the tissue degeneration, the deformation data were blended into the 3D anatomy volume. The final design can be seen in **Figure 1**.

2.2.2. Importance-driven volume rendering design (IDVR design)

In comparison to the overview design, the IDVR design offered a more specialized view for a detailed examination of specific brain areas. The anatomy was visualized in an opaque fashion and a cutout facilitated a view into the volume by removing only as much anatomy as necessary and staying automatically aligned to the user's view direction. For this purpose, the design used an advanced algorithm based on Viola et al. (2004), so that the specified VOI was always visible due to a view-dependent cutout. The original work defines a conical cutout, with the tip of the cone being determined by the VOI's deepest voxel along the view direction. The faces of the cone help to determine the depth of the VOI in the overall volume. However, a drawback of this previous algorithm was, nearby structures at the same depth of the VOI may be covered by the surrounding brain tissue. Therefore, it was more favorable to expand the cutout by using a section-like plane that is aligned with the back side (in viewing direction) of the VOI. Thus, Viola's algorithm was modified by using a frustum shaped cutout instead of a conical one. The top plane of this frustum was positioned at the level of the deepest VOI voxel. In the present study, the VOI was defined by a neuroanatomical region. Neuroanatomical regions, e.g., cortical areas or nuclei, have often a complex structure, so that its depth texture is strongly varying. Hence, creating the section only on basis of one depth value would neglect nearby structures on all other depth levels of the VOI. Therefore, we adapted the algorithm to define the top surface of a frustum-like cutout with a section that is approximated using all values of the VOI's backface (cf. **Figure 2**).

The cutout calculation was based on multi-pass raycast rendering and worked as follows. In the first pass, a special modification of a depth texture of the VOI was defined as not only the depth values are interesting, but also the exact sample position in the volume. Therefore, rays were directed into the volume that were defined by a previously calculated ray entry points texture T_R and a ray exit point texture T_E. Along each ray, the x-, y-, and z-coordinate of the deepest VOI voxel and the accumulated length l_a until this point were determined. These values were stored into the output texture T_V of this first rendering pass. If the ray did not hit the VOI at all, the four texture element (texel) values were set to zero.

In the second rendering pass, the cutout was defined. To find the best definition of the top surface of the frustum with respect to the best information retrieval and smoothness, three different implementations were tested. The first two approaches of the top surface definition varied only in the determination of the texel $P_V \in T_V$ that is used as reference point for further calculations (cf. **Figure 4** left, middle). In the first case a vector $\overrightarrow{P_R P_{V_1}}$ was sought, where $P_R \in T_R$ was the current ray entry point and $P_{V_1} \in T_V$ is defined by the closest texel of T_V with $l_a > 0$, within a maximum distance d in X- and Y-direction of T_V. Therefore, the algorithm iterated over all texels of T_V from $-d$ to $+d$ distance in X- and Y-direction starting from the texel with the same texel coordinates as P_R.

In the second case, we adapted this iteration step by not minimizing $|\overrightarrow{P_R P_{V_1}}|$, but rather find the texel $P_{V_2} \in T_V$ that created a vector $\overrightarrow{P_R P_{V_2}}$ with a minimum angle between $\overrightarrow{P_R P_{V_2}}$ and the X- or Y-axis. The iteration starts at 0, checks in $\pm d$ in X- and Y-direction, and terminates if a sufficient P_{V_2} is found. From this point on, the calculations were identical for the first and second implementation and we defined $P_V = P_{V_1}$ or $P_V = P_{V_2}$, respectively.

Let r_1 be the maximum length of $\overrightarrow{P_R P_V}$, with

$$r_1 = \sqrt{d^2 + d^2}, \text{ where } \sqrt{d^2 + d^2} \geq |\overrightarrow{P_R P_V}|. \quad (1)$$

If $\overrightarrow{P_R P_V}$ existed, it was possible to determine a vector $\overrightarrow{RV}$, with V being the corresponding voxel of the VOI to P_V saved in the output texture of the first rendering pass T_V, and R being defined with the help of the congruence theorem of triangles, where an edge with an angle of 90° could be constructed from the view ray to V (cf. **Figure 5**). If $|\overrightarrow{RV}|$ was within a radius r_2, with $r_2 \leq r_1$, the ray hit the top surface of the frustum and the depth value c of

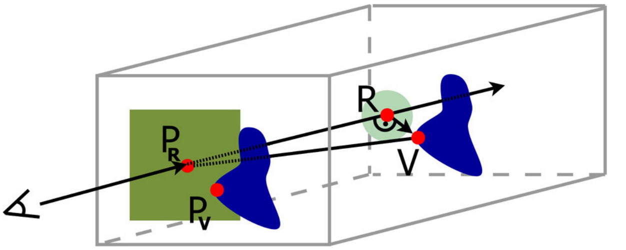


FIGURE 5 | Schematic illustration of the construction from the closest depth point of the volume of interest *V* onto the view ray *R*. In dark blue we see in the back the volume of interest and its projected depth texture on the near clipping plane. The dark green area limits our search area from the ray entry point P_R to a nearby VOI point P_V, and the light green circle with radius r_2 limits the top surface size.

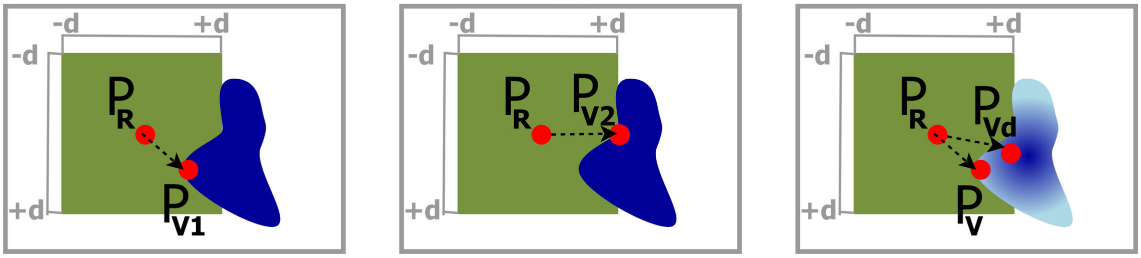


FIGURE 4 | Determination of the local depth value in the cutout. Left: Use depth of P_{V_1} as the closest texel of the VOI's depth texture to the ray entry point P_R. **Middle:** P_{V_2} is the most straight aligned texel in relation to P_R. **Right:** A darker color in the VOI depicts a higher depth value. P_V is the closest texel to P_R and is used to calculate the distance to the VOI, but P_{V_d} has the highest depth value in distance d around P_R, and is utilized as depth value.

the cutout for the current view ray was set to the depth value of V. Otherwise, the cutout depth c was calculated as follows

$$c = |\overrightarrow{RV}| - \frac{|\overrightarrow{P_R P_V}| - r_2}{r_1 - r_2}. \tag{2}$$

The result of the first approach showed circular artifacts around small parts of the VOI that stuck out, and where depth changes of the VOI occurred (cf. **Figure 6** left). For the second approach, we see hard edges in diagonal orientation (cf. **Figure 6** middle). To create a smoother frustum top surface, neglecting small outliers, we implemented a third approach which is schematically shown in **Figure 4** right. In addition to P_{V_1} the texel P_{V_d} in T_V was sought within a maximum distance $\pm d$ to P_R in X- and Y-direction with the highest l_a value. $\overrightarrow{P_R P_V}$ was calculated as in the previous approaches, but $\overrightarrow{RV}$ was replaced with $\overrightarrow{RV_d}$ and the additional depth had to be included in the calculation of the frustum faces, resulting in

$$c = |\overrightarrow{RV_d}| - |\overrightarrow{RV_d}| \cdot \frac{\left(|\overrightarrow{P_R P_V}| - r_2\right)}{r_1 - r_2}. \tag{3}$$

Although the depth value of the top surface is determined by the depth value of V_d, $|\overrightarrow{RV}|$ still defines whether the view ray hits the top surface or is part of a frustum face. An exemplary smoothed cutout can be seen in **Figure 6** right.

2.3. INTERACTION

Depth perception can be improved by rotating, panning, and zooming (cf. Swanston and Gogel, 1986), suggesting that interaction with the brain model in the 3D visualization is desirable. In immersive virtual environments, correct depth relations can already be perceived without additional intentional interaction. Therefore, we provided the application for both, standard desktop setups and 3D immersive virtual environments with stereoscopic vision. To this end, we used the open source, cross-platform ViSTA toolkit (cf. Assenmacher and Kuhlen, 2008) for easy scalability to different systems. In the immersive setup, the depth impression of the cutout in the IDVR design became better comprehensible and the location of the border between faces and top surface of the frustum was clearly visible. The advantage of virtual environments over 2D displays for depth perception and estimation were shown in several studies, e.g., in Armbrüster et al. (2006) and Naceri et al. (2010) and in particular for volume rendered data in Laha et al. (2012).

Because the application was provided for Virtual Reality setups, an alternative to the classical 2D menu interaction became necessary. To this end, we decided to use extended pie menus as described by Gebhardt et al. (2013). They scale to 2D and 3D environments and can interactively be moved in the scene while staying aligned to the user's orientation. The menu is hierarchically arranged and can be divided into various submenus (cf. **Figure 3**). The most important interactions that were controlled via these menus are explained below.

Time Navigation–This submenu held the time navigation, where the user was able to set the animation speed or can manually step through all time steps of the presented data.

Cortical Areas–Predefined anatomical regions such as cytoarchitectonic areas from the JuBrain atlas could be selected for visualization by this submenu. Color codes of these regions and their arrangement in groups could be defined in a separate settings file that was read when the program starts. These definitions were also represented in the pie menu, allowing the user to hide or display whole area groups with a single mouse click. Furthermore, the opacity could be adjusted to provide a view onto degeneration occurring inside the anatomical regions.

Importance Driven Volume Rendering–Here the user could switch to the IDVR design. Depending on the available data sets of the subject, a variety of options existed. By default the VOI was defined by the visible anatomical regions. If provided the user was able to select any other VOI defined by field data as well. It can be useful to show other data next to the VOI in the cutout. Therefore, the user might choose to visualize the degeneration, or if provided, any other data exempt from the anatomy.

Color Map–This submenu allowed users to adapt the opacity and enhance the contrast of the anatomy. The contrast and opacity parameters have to be adapted to the range of voxel values of the data sets which are to be visualized. Furthermore, the color menu allowed users to change the threshold for the deformation values to exclude small degeneration values and set the focus on larger ones. The adjustability of the opacity for these values led to a good spatial orientation particularly in the IDVR design because the degeneration could be visualized in

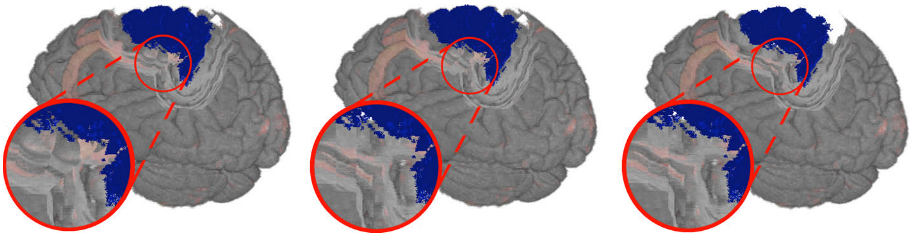


FIGURE 6 | Clipping artifacts depend on the definition of the distance value and are clearly visible when observing transitions in the sulci (dark gray) in the detail view. Left: Circular artifacts when using the closest voxel of the VOI. **Middle:** Diagonal artifacts when preferring voxels in straight alignment. **Right:** Smoothest result with homogeneous depth values for a nearby voxel.

the cutout and a high transparency preserves a good view onto the cutout faces. Moreover, in the overview design the right balance between opacity of cortical areas and degeneration allowed for good visual comparability of relationship between the two volumes.

3. RESULTS AND DISCUSSION

The visualization of anatomical structure and superimposed field data by volume rendering enables neuroscientists to observe the data described in section 2.1 not only on the surface, but also inside the brain, and to get a better impression about the spatial extent of regional volume loss in the context of the individual brain anatomy. This is an advantage over visualization based on surface reconstruction, because the latter is typically limited to field data on or near the surface, which causes a great loss of information. In studies of neurodegenerative diseases, it makes an important difference if the tissue atrophy that is quantified by the superimposed volume change data occurs only in the cortex, or if also subcortical regions and white matter (i.e., fibers deep inside the brain which connect brain regions) are affected. For instance, the data examined here show a progressive atrophy, which includes both the motor cortex areas and the pyramidal tract. But transparent volume rendering of the brain often yields diffuse borders, whereas in surface based visualization the perception of the surface shape can be enhanced, for example, by the simulated effects of lighting, reflection, and shadows. Moreover, brain structures that are deep inside the brain and have only low contrast to their environment are hardly perceived when volume rendering of an MRT image is used exclusively.

A 2D section is added in the overview design onto which the voxel values of the structural image of the brain are mapped, so that cortex, subcortical nuclei, and white matter can easily be identified according to existing brain atlases. The brain in front of the added plane is not removed and thus landmarks for an anatomical localization are still provided. This design is particularly useful, when certain features near or within the brain cortex like the atrophy of certain gyri shall be shown.

In the IDVR design the brain in front of the clipping surface is removed, whereas the anatomical maps and the volume change data are still completely rendered. This design is more appropriate to show deeper parts of the brain: By means of rendering the anatomical regions in front of the clipping surface and the surface texture, a good localization is possible. In particular, the overlap of anatomical regions and volume change data is intuitively displayed by the blending of their colors (cf. **Figure 7**).

Moreover, the automatic alignment of the clipping surface when the brain is moved relative to the observer enables a simple interaction, which gives a good spatial perception even in non-virtual environments. The extent of the removed part of the brain can be controlled by the selection of predefined anatomical regions and additional parameters like the aperture of the frustum. We observed that using the application in a user-friendly immersive virtual environment enhances the perception of the spatial relations, in particular of spatial depth.

Since this application is to be used interactively, the frame rate has an important influence on the performance, which is why it was investigated in more detail. In comparison to visualizations of geometries, volume rendering approaches lack in performance. Moreover, the use of stereo viewing in an immersive virtual environment halves the frame rate due to the generation of two simultaneous images (one per eye). We tested our application on two systems: The first is used as desktop environment, and runs Windows 7 on Intel Xeon CPU E5540 with four cores at 2.5 GHz, an Nvidia GeForce GTX 480 graphics card, and 12 GB RAM; the second system is used as virtual environment with a passive stereo system, head tracking, and runs Windows 7 on Intel Xeon CPU E5530 with four cores at 2.4 GHz, but with an Nvidia Quadro 6000 graphics card, and 4 GB of RAM. With a resolution of 1400 × 1050 we achieve for the overview design about 30 frames per second (fps) in stereo mode which fulfills the requirement for interactivity. The implementation of the IDVR design is based on the same volume renderer, thus a higher frame rate cannot be expected. We are limited by the iteration over all neighboring texels when seeking P_V which it cannot be early terminated, because there is always the possibility to find a closer V, respectively, a V_d with higher l_a. To solve this problem, we introduced a parameter to the IDVR calculation to set the accuracy in the iteration. Assuming that the resolution of the texture T_V is sufficiently large, every i-th texel can be skipped and is not tested to be a candidate for V or V_d. The value for i can be changed via the pie menu. For $i = 2$ nearly no visual artifacts can be found. For $i = 3$ noticeable impacts in form of loose wrong depth values appear, but this method is still reasonable, because it allows for better performance and the artifacts are not disturbing the overall depth perception in the cutout. The achieved frame rates (cf. **Table 1**) for a scene comparable to the one shown in **Figure 3** show that also the IDVR design can be used interactively in virtual environments, but clearly needs performance improvements. However, in general the frame rate of

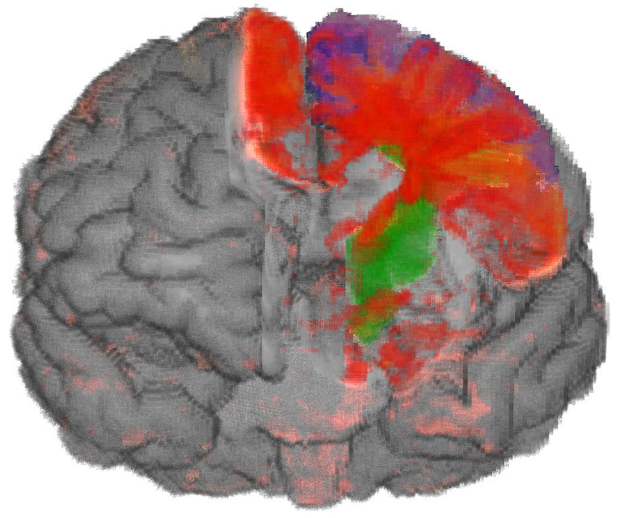


FIGURE 7 | Atrophic part of the brain (red to yellow) of a person with CBS and maps of anatomical regions (blue premotor cortex, green cortico-spinal tract). The removed part of the brain is adjusted to the selected anatomical regions. The overlap between atrophic parts and anatomical regions can be recognized by the blending of different colors.

Table 1 | Results of performance tests of the visualization designs (averaged values from different view points), the overview design with common volume rendering and the IDVR design with different precision states for the cutout generation.

	Nvidia GeForce GTX 480 (mono view)	Nvidia Quadro 6000 (stereo view)
Overview Design	55 fps	30 fps
IDVR Design		
$i = 1$	4 fps	3 fps
$i = 2$	11 fps	7 fps
$i = 3$	18 fps	10 fps

our application is highly dependent on the window size, the size of the volume on the screen, and for the IDVR design on the value of distance *d*.

4. CONCLUSION AND FUTURE WORK

In this paper, we have introduced two visualization designs to address the challenge of depicting complex 3D information of human brain data. Whereas the first approach provides an overview of the data, the second allows for a more detailed examination. Current work as, for example, Laha et al. (2012) already showed that the visualization in virtual environments supports the analysis of volume data, but more studies are necessary in this field. First, the general improvement of spatial impression and the ability of correct spatial localization with our designs in comparison to commonly used 2D section views should be proven, for example, by determining the extension of an artificial tissue degeneration and its spatial localization. Second, this experiment should be repeated in an immersive virtual environment and then compared to the results of desktop environments. Moreover, since our application is supposed to benefit the daily workflow of neuroscientists, their expert impression of additional or more easily grasped information should be gathered.

Illustrations in anatomical text books like Nieuwenhuys et al. (2008) are excellent artworks that selectively emphasize certain structural entities or parts of the brain while showing the surrounding brain structure. These figures were artistic drawings, but it is desirable to achieve similar presentations by computerized 3D visualizations which can be manipulated by user interaction (cf. Bruckner et al., 2006). In particular, the gradient based emphasis of surface structures could be used to stress the brain surface and show the ventricles in the overview design more clearly. Furthermore, an enhanced contrast-to-noise ratio of the MRI data and a visual smoothing would improve the quality of the visualization and allow for easier analysis. As discussed in section 3, the IDVR design could benefit from a faster cutout calculation to ensure a higher frame rate which in turn would lead to increased interactivity. Therefore, one approach might be to use distance maps created from the result of the first rendering pass and discard the iteration approach during the second rendering pass that is mainly responsible for the frame rate decrease.

In conclusion, we have significantly improved the spatial localization of brain structures affected by CBS and the understanding of its temporal progression which motivates further research, and an application to other neurological and psychiatric disorders.

ACKNOWLEDGMENTS

This work was funded by the Helmholtz Portfolio Theme "Supercomputing and Modeling for the Human Brain" and partially by the "Human Brain Project" as part of the FET Flagship Programme.

REFERENCES

Amunts, K., Schleicher, A., and Zilles, K. (2007). Cytoarchitecture of the cerebral cortex–more than localization. *Neuroimage* 37, 1061–1065. doi: 10.1016/j.neuroimage.2007.02.037

Armbrüster, C., Wolter, M., Valvoda, J. T., Kuhlen, T., Spijkers, W., and Fimm, B. (2006). Virtual reality as a research tool in neuropsychology: depth estimations in the peripersonal space. *CyberPsychol. Behav.* 9, 654. doi: 10.1089/cpb.2006.9.653

Assenmacher, I., and Kuhlen, T. (2008). "The ViSTA virtual reality toolkit," in *Proceedings of the IEEE VR 2008 Workshop Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, (Reno, Nevada: Shaker Verlag), 23–28.

Born, S., Jainek, W., Hlawitschka, M., Scheuermann, G., Trantakis, C., Meixensberger, J., et al. (2009). "Multimodal visualization of DTI and fMRI data using illustrative methods," in *Bildverarbeitung Für Die Medizin 2009*, eds H.-P. Meinzer, T. Deserno, H. Handels, and T. Tolxdorff (Berlin; Heidelberg: Springer, Informatik aktuell), 6–10. doi: 10.1007/978-3-540-93860-6_2

Bruckner, S., Grimm, S., Kanitsar, A., and Gröller, M. E. (2006). Illustrative context-preserving exploration of volume data. *IEEE Trans. Visual. Comput. Graph.* 12, 1559–1569. doi: 10.1109/TVCG.2006.96

Cabral, B., Cam, N., and Foran, J. (1994). "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," in *Proceedings of the 1994 Symposium on Volume Visualization, VVS '94*, (Tysons Corner, Virginia: ACM), 91–98. doi: 10.1145/197938.197972

Gebhardt, S., Pick, S., Leithold, F., Hentschel, B., and Kuhlen, T. (2013). Extended pie menus for immersive virtual environments. *IEEE Trans. Visual. Comput. Graph.* 19, 644–651. doi: 10.1109/TVCG.2013.31

Hauser, H., Mroz, L., Bischi, G. I., and Gröller, M. E. (2001). Two-level volume rendering. *IEEE Trans. Visual. Comput. Graph.* 7, 242–252. doi: 10.1109/2945.942692

Krüger, J., Schneider, J., and Westermann, R. (2006). ClearView: an interactive context preserving hotspot visualization technique. *IEEE Trans. Visual. Comput. Graph.* 12, 941–948. doi: 10.1109/TVCG.2006.124

Laha, B., Sensharma, K., Schiffbauer, J., and Bowman, D. (2012). Effects of immersion on visual analysis of volume data. *IEEE Trans. Visual. Comput. Graph.* 18, 597–606. doi: 10.1109/TVCG.2012.42

Naceri, A., Chellali, R., Dionnet, F., and Toma, S. (2010). Depth perception within virtual environments: comparison between two display technologies. *Int. J. Adv. Intel. Syst.* 3, 51–64.

Nieuwenhuys, R., Voogd, J., and van Huijzen, C. (2008). *The Human Central Nervous System, 4th Edn.* Berlin: Springer-Verlag.

Olabi, B., Ellison-Wright, I., Bullmore, E., and Lawrie, S. (2012). Structural brain changes in first episode Schizophrenia compared with fronto-temporal lobar degeneration: a meta-analysis. *BMC Psychiatry* 12:1–13. doi: 10.1186/1471-244X-12-104

Pieperhoff, P., Südmeyer, M., Hömke, L., Zilles, K., Schnitzler, A., and Amunts, K. (2008). Detection of structural changes of the human brain in longitudinally acquired MR images by deformation field morphometry: methodological analysis, validation and application. *Neuroimage* 43, 269–287. doi: 10.1016/j.neuroimage.2008.07.031

Rick, T., von Kapri, A., Caspers, S., Amunts, K., Zilles, K., and Kuhlen, T. (2011). Visualization of probabilistic fiber tracts in virtual reality. *Stud. Health Technol. Inform.* 163, 486–492. doi: 10.3233/978-1-60750-706-2-486

Rieder, C., Brachmann, C., Hofmann, B., Klein, J., Köhn, A., Ojdanic, D., et al. (2013). "Interactive visualization of neuroanatomical data for a hands-on multimedia exhibit," in *Visualization in Medicine and Life Sciences*, eds L. Linsen, H. C. Hege, and B. Hamann (Leipzig: Eurographics Association), 37–41.

Rößler, F., Tejada, E., Fangmeier, T., Ertl, T., and Knauff, M. (2006). "GPU-based multi-volume rendering for the visualization of functional brain images," in *Proceedings of SIMVIS '06* (Magdeburg: Publishing House), 305–318.

Südmeyer, M., Pieperhoff, P., Ferrea, S., Krause, H., Groiss, S., Elben, S., et al. (2012). Longitudinal deformation-based morphometry reveals spatio-temporal dynamics of brain volume changes in patients with corticobasal syndrome. *PLoS ONE* 7:e41873. doi: 10.1371/journal.pone.0041873

Swanston, M. T., and Gogel, W. C. (1986). Perceived size and motion in depth from optical expansion. *Atten. Percept. Psychophys.* 39, 309–326. doi: 10.3758/BF03202998

Thompson, P. M., Hayashi, K. M., Dutton, R. A., Chiang, M.-C., Leow, A. D., Sowell, E. R., et al. (2007). Tracking Alzheimer's disease. *Ann. N.Y. Acad. Sci.* 1097, 183–214. doi: 10.1196/annals.1379.017

Viola, I., Kanitsar, A., and Gröller, M. E. (2004). "Importance-driven volume rendering," in *Proceedings of IEEE Visualization '04* (Washington, DC: IEEE Computer Society), 139–145.

Weber, S., Habel, U., Amunts, K., and Schneider, F. (2008). Structural brain abnormalities in psychopaths - a review. *Behav. Sci. Law* 26, 7–28. doi: 10.1002/bsl.802

Zhou, Y., Kierans, A., Kenul, D., Ge, Y., Rath, J., Reaume, J., et al. (2013). Mild traumatic brain injury: longitudinal regional brain volume changes. *Radiology* 267, 880–890. doi: 10.1148/radiol.13122542

Zilles, K., Schleicher, A., Palomero-Gallagher, N., and Amunts, K. (2002). *Quantitative Analysis of Cyto- and Receptor Architecture of The Human Brain, Chapter 21, 2nd Edn.* (San Diego: Elsevier), 573–602.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 19 December 2013; paper pending published: 09 February 2014; accepted: 28 March 2014; published online: 05 May 2014.

Citation: Hänel C, Pieperhoff P, Hentschel B, Amunts K and Kuhlen T (2014) Interactive 3D visualization of structural changes in the brain of a person with corticobasal syndrome. Front. Neuroinform. ***8****:42. doi: 10.3389/fninf.2014.00042*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 22 August 2013
doi: 10.3389/fninf.2013.00016

Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness

Adrian Andronache[1], Cristina Rosazza[1,2], Davide Sattin[3], Matilde Leonardi[3], Ludovico D'Incerti[1] and Ludovico Minati[2]* on behalf of the Coma Research Centre (CRC) – Besta Institute

[1] Neuroradiology Unit, Fondazione IRCCS Istituto Neurologico "Carlo Besta", Milan, Italy
[2] Scientific Department, Fondazione IRCCS Istituto Neurologico "Carlo Besta", Milan, Italy
[3] Neurology, Public Health, Disability Unit, Scientific Department, Fondazione IRCCS Istituto Neurologico "Carlo Besta", Milan, Italy

Edited by:
Daniel Marcus, Washington University in St. Louis, USA

Reviewed by:
J. Anthony Wilson, Washington University School of Medicine, USA

****Correspondence:***
Ludovico Minati, Scientific Department, Fondazione IRCCS Istituto Neurologico "Carlo Besta", Via Celoria 11, 20133 Milan, Italy
e-mail: lminati@istituto-besta.it

An emerging application of resting-state functional MRI (rs-fMRI) is the study of patients with disorders of consciousness (DoC), where integrity of default-mode network (DMN) activity is associated to the clinical level of preservation of consciousness. Due to the inherent inability to follow verbal instructions, arousal induced by scanning noise and postural pain, these patients tend to exhibit substantial levels of movement. This results in spurious, non-neural fluctuations of the rs-fMRI signal, which impair the evaluation of residual functional connectivity. Here, the effect of data preprocessing choices on the detectability of the DMN was systematically evaluated in a representative cohort of 30 clinically and etiologically heterogeneous DoC patients and 33 healthy controls. Starting from a standard preprocessing pipeline, additional steps were gradually inserted, namely band-pass filtering (BPF), removal of co-variance with the movement vectors, removal of co-variance with the global brain parenchyma signal, rejection of realignment outlier volumes and ventricle masking. Both independent-component analysis (ICA) and seed-based analysis (SBA) were performed, and DMN detectability was assessed quantitatively as well as visually. The results of the present study strongly show that the detection of DMN activity in the sub-optimal fMRI series acquired on DoC patients is contingent on the use of adequate filtering steps. ICA and SBA are differently affected but give convergent findings for high-grade preprocessing. We propose that future studies in this area should adopt the described preprocessing procedures as a minimum standard to reduce the probability of wrongly inferring that DMN activity is absent.

Keywords: functional MRI (fMRI), resting-state, functional connectivity, disorders of consciousness, vegetative state, minimally-conscious state, data preprocessing

INTRODUCTION

In recent years, resting-state functional MRI (rs-fMRI) has attracted substantial research and clinical interest. In contrast with fMRI based on active tasks, it is a straightforward form of functional imaging suitable for the study of patients who are unable to follow procedural instructions or are generally unresponsive. Alongside practical considerations, there is increased awareness of the importance of intrinsic brain activity in supporting behavioral function and determining metabolism (e.g., Fox and Raichle, 2007; Rosazza and Minati, 2011).

An emerging application of rs-fMRI is the study of patients with disorders of consciousness (DoC), an etiologically heterogeneous condition that typically follows substantial brain damage due to vascular, hypoxic or traumatic insults. The clinical phenotype is highly variable, ranging from complete absence of wilful responses (vegetative state) to situations where awareness is fluctuating and a rudimentary communication code may be established (minimally-conscious state or severe disability; e.g., Laureys, 2005; Owen and Coleman, 2008).

In the healthy brain, rs-fMRI reveals a set of well-reproducible, separable activity components which appear to correlate with specific sensory, motor and cognitive functions (Biswal et al., 2010; Allen et al., 2011). In particular, the default-mode network (DMN) has received considerable attention as a potential proxy of large-scale integrative processes related to awareness, interoception and memory consolidation. This bi-hemispheric network has its main constituent nodes in the precuneus, lateral parietal cortex and medial prefrontal cortex, and exhibits a well-reproducible, graded response to wakefulness, sleep and coma (Raichle et al., 2001; Buckner et al., 2008; Rosazza and Minati, 2011).

The severity of the clinical phenotype of patients in vegetative state or minimally conscious state is reflected in the level of residual functional connectivity across the DMN (Boly et al., 2009; Cauda et al., 2009; Vanhaudenhuyse et al., 2010; Soddu et al., 2011) and other networks (Owen et al., 2006; Owen and Coleman, 2007), with recent work also indicating specific alterations in the relationships across networks, particularly between the DMN and the fronto-parietal component (Boly et al., 2009; Noirhomme et al., 2010). Importantly, the intensity of connectivity across the DMN nodes and, consequentially, the detectability of the network as a whole appear to be coupled to the level of residual consciousness, as assessed by established clinical scales (Vanhaudenhuyse et al., 2010). Rs-fMRI is therefore of particular relevance for the study of DoC patients, since it can help to determine how large-scale integrative processes are affected in the presence of impaired consciousness.

A major challenge for the use of rs-fMRI in clinical populations is head movement, consequential to several factors including the inability for the patient to understand and comply with verbal instructions, emotional arousal caused by the scanner environment, decorticate or decerebrate posture and postural pain. Even when gross imaging artifacts are absent owing to the rapidity of echo-planar acquisition and time-series volumes are accurately realigned, significant signal modulations are introduced by movement due to multiple factors including inhomogeneous coil sensitivity, inhomogeneous coil loading by the head, interaction between susceptibility gradients and head movement, and partial-volume effects. Such contaminations can introduce spurious correlations as well as mask coherent neuronal sources of blood-oxygen level-dependent (BOLD) signal fluctuations, making it impossible to draw reliable inferences on the degree of preservation of functional connectivity (Friston et al., 1996; Hutton et al., 2002; Johnstone et al., 2006; Strother, 2006; Power et al., 2012).

Thus, head movement represents a particularly insidious confound for the study of patients with DoC (Giacino et al., 2006; Owen and Coleman, 2007; Soddu et al., 2011) because very large variability of residual neuronal function is expected ab-initio, as testified by the fact that the EEG can range from near-normal to near-isoelectric (Soddu et al., 2012), and operators may therefore be inclined to accept the findings of rs-fMRI uncritically.

Pre-processing techniques to remove physiological noise and movement artifacts in rs-fMRI have been investigated extensively in healthy participants with reference to both data-driven (i.e., independent-component analysis, ICA) and anatomy-driven (i.e., seed-based analysis, SBA) analyses (Birn et al., 2006; Lund et al., 2006; Fox et al., 2009; Murphy et al., 2009; Weissenbacher et al., 2009; Van Dijk et al., 2010). Existing studies have demonstrated the importance of removing by linear regression co-variance with movement parameters (Power et al., 2012; Van Dijk et al., 2012) and physiological variables (Corfield et al., 2001; Birn et al., 2006; Weissenbacher et al., 2009), either measured directly or inferred from the rs-fMRI time-series. Several reports have also underlined the utility of removing diffuse and un-specific signal fluctuations, indexed by averaging signal over the whole brain: while this may induce artifactual anti-correlations, it strongly limits the effect of unaccounted sources of global noise over inter-regional correlation estimates (Desjardins et al., 2001; Macey et al., 2004; Murphy et al., 2009). Further, it has been demonstrated that the confounding effect of movement can be attenuated by combining regression of the movement parameters with the exclusion and replacement by interpolation of selected contaminated volumes; this approach is particularly appropriate in the presence of brief, large movements (Carp, 2013). In recent work on healthy controls, the effect of the available filtering techniques was systematically investigated, and it was concluded that consideration of the parameters listed above alongside the first temporal derivative of movement enhances the sensitivity and stability of connectivity inferences (Van Dijk et al., 2012; Satterthwaite et al., 2013). There is, however, a lack of consistency in terms of preprocessing methods across the existing rs-fMRI investigations of residual neural function in DoC: while in some studies movement-related, physiological and unspecific global BOLD signal variance were explicitly removed, in others more basic data-preprocessing chains were utilized. In particular, none of studies the authors are aware of have included specific preprocessing steps to reduce the impact of the large, sudden movements and substantial gross anatomical damage present in this population, e.g., by outlier rejection and masking (Boly et al., 2009; Cauda et al., 2009; Vanhaudenhuyse et al., 2010; Soddu et al., 2011, 2012).

A crucial and unresolved question pertains to what extent the large variability observed in this clinical group truly represents neural differences rather than being consequential to movement and other confounds. From a methodological viewpoint, there is a need for a systematic evaluation of the effect of preprocessing choices on data from this specific clinical population, and for clear guidelines on how to best preprocess the rs-fMRI datasets acquired for diagnostic and research purposes, to ensure the best yield in terms of detectability of residual DMN function.

Here, we comprehensively investigated how inserting specific filtering steps in the preprocessing chain can improve the detectability of the DMN or its residual portions in a clinically and aetiologically heterogeneous population of DoC patients. We hypothesized that using a tailored preprocessing chain would substantially improve DMN detectability, as revealed by automated measurements as well as qualitative assessments. Since ICA and SBA are often interchangeably utilized, in spite of their substantially different computational properties (e.g., Rosazza et al., 2012), we also investigated whether the two techniques are differently sensitive to data preprocessing.

METHODS

PARTICIPANTS

All investigational protocols were approved by the institutional ethics committee and written informed consent was always obtained from the healthy participants and the legal representative of the patients. The study was conducted on 30 consecutive patients with a clinical diagnosis of vegetative state or minimally conscious state and 33 healthy volunteers. The selection criterion for the patients was the detectability of the DMN with at least one data analysis technique in one of the 5 different procedures; patients in whom in the DMN appeared completely absent, irrespective of data preprocessing and analysis choices, were excluded a-priori, since the purpose of the present study was to demonstrate the differential effect of data preprocessing choices on DMN detectability. In the recruitment period, 25 other patients were scanned, but rejected as the DMN was not detectable with either ICA or SBA, irrespective of preprocessing.

The average patient age was 54 years (range 22–82), average disease duration was 34 months (range 7–105), 13 patients were female; regarding etiology, 13 have had head trauma, 11 intracranial hemorrhage and 6 cerebral anoxia. For controls, the average age was 39 years (range 17–66). All patients were assessed and evaluated with the Coma Recovery Scale-Revised (CRS-R; Giacino et al., 2004; Lombardi et al., 2007) and with the Coma Near-Coma scale (CNC; Rappaport, 2005). According to internationally accepted criteria (Multi-Society, 1994; Giacino, 2004), 15 patients were diagnosed as being in vegetative state (CRS-R 6.4 ± 1.8, CNC 2.4 ± 0.5) with the remaining 15 being in minimally conscious state (CRS-R 13.7 ± 5.4, CNC 1.3 ± 0.6).

DATA ACQUISITION

Functional imaging was performed on a 3 Tesla scanner equipped with a 32-channel head coil (Achieva, Philips Healthcare BV, Best, NL). Two hundred functional volumes were acquired by means of an axial gradient-echo echo-planar sequence, having $TR = 2800$ ms, $TE = 30$ ms, $\alpha = 70°$, 2.5 mm isotropic voxel size, 90 × 95 matrix size, 50 slices with 10% gap, ascending order. Sequence duration was ~9.5 min. When possible given the patient posture, the head was gently restrained using foam pillows, and a wedge was positioned under the knees to minimize spine movement. The relatively small voxel size was chosen primarily to reduce spatial distortions in the presence of inhomogeneous susceptibility due to macroscopic lesions and deposits.

DATA PREPROCESSING

Five data preprocessing procedures of increasing complexity, consisting of different combinations of standard modules implemented in SPM8 (Wellcome Trust Centre for Neuroimaging, London, UK) and custom code developed in MatLab 7 (Mathworks Inc., Natick MA, USA), were compared (**Figure 1**).

Procedure 1 (P1) consisted of the standard SPM8 workflow for fMRI: rigid-body realignment to average volume with minimization of squared differences (R), slice-timing correction (ST), normalization to MNI space by co-registration to the individual T_1 structural scan and subsequent segmentation (N), and spatial smoothing using an isotropic Gaussian kernel having FWHM 8 mm (S). The absence of gross normalization errors was visually confirmed by an experienced operator for all patients.

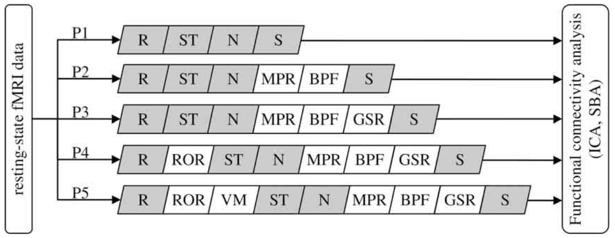


FIGURE 1 | Definition of the data preprocessing pipeline for the five procedures (P1–P5) under comparison. R, realignment; ST, slice-timing correction; N, normalization to MNI space; S, spatial smoothing; MPR, removal of co-variance with movement parameters; BPF, band-pass filtering; GSR, removal of global parenchymal signal; ROR, removal of realignment outliers; VM, ventricle masking. The modules in gray are standard SPM8 functions, the others are functions developed in-house (see text).

Procedure 2 (P2) additionally included masking with a standard brain-mask to remove all voxels outside the brain outline (but not the ventricles), removal of movement-related variance (MPR) and band-pass filtering (BPF). Movement-related variance was removed by multilinear regression of the individual voxel time-courses with respect to the six movement vectors, measured in absolute terms with respect to the first volume. BPF was performed removing baseline fluctuations, e.g., related to gradient system heating, by fitting and subtracting a 3rd order polynomial, followed by low-pass filtering with a Butterworth filter of order 1 having $f_{-3dB} = 0.1$ Hz and applied twice in opposite directions to attenuate rapid, non-neural BOLD signal fluctuations (e.g., cardiac pulsatility).

Procedure 3 (P3) added global signal regression (GSR), i.e., the removal by linear regression of the variance correlated to average signal intensity time-course calculated over all voxels included in the brain parenchyma mask, derived from SPM segmentation. Performing this operation is advised in Weissenbacher et al. (2009) and Van Dijk et al. (2010) as it attenuates topographically-unspecific temporal variance, e.g., related to residual baseline instability effects and systemic sources of physiological noise, which can positively bias connectivity inferences. While this step is generally deemed not necessary for ICA, here a common set of preprocessing pipelines was considered and ICA/SBA were therefore performed on the same data, including the GSR step.

Procedure 4 (P4) added the removal of realignment outliers (ROR), i.e., the identification and replacement of volumes having large residual mean-square difference from the average volume after realignment, indicating the presence of macroscopic imaging artifacts due to head movement, similarly to the work of Carp (2013). Consideration of mean-square signal difference enables a more direct assessment of signal contamination with respect to distance from reference volume; for example, in presence of sudden movements this criterion promptly identifies volumes affected by "shear" between the first and last sections: for these, the translation/rotation realignment parameters may not differ substantially from the neighboring volumes but the attained overlap with the reference volume is poor due to distortion. Volumes having residual mean-square difference larger than 1.5 times the interquartile range calculated over all volumes of a series were considered potential outliers. However, outlier rejection was actually performed only if the mean-square difference exceeded a reference value, empirically set to 10% of the average of the three mean-square differences obtained by artificially displacing by one voxel along the three axes the image used as reference in the realignment process. When less than 10 consecutive outliers were identified, to avoid introducing temporal discontinuities they were replaced with the multilinear interpolation of the nearest preserved volumes; groups of more than 10 consecutive outliers were removed altogether.

Procedure 5 (P5) additionally included masking to remove the ventricles (VM) and was motivated by the observation of very large ventricles with substantial flow-induced signal fluctuations in some patients. For each scan, the ventricles were identified by average signal intensity thresholding followed by morphological filtering to remove speckles, fill holes and identify the connected-component representing the ventricles.

Removal of voxels outside the brain outline (P2) and in the ventricles (P5) was deemed relevant here because it excludes a range of non-neural signal sources such as pulsating cerebrospinal fluid and eye movements; while in healthy participants these do not impact ICA substantially, it was hypothesized that in patients their removal might facilitate the proper un-mixing of weak neural signals, especially given that substantial brain atrophy can be present and the relative representation of cerebrospinal fluid voxels can be substantially higher with respect to controls.

DATA ANALYSIS

ICA was performed independently for each participant, using the group ICA of fMRI toolbox (GIFT, MIALab, University of New Mexico, USA) and assuming a fixed number of 20 independent components (Calhoun et al., 2001, 2008). The component corresponding to DMN activity was identified upon agreement of two experienced observers, who searched for significant correlation clusters (at $z > 2$) specifically in the precuneus (PCC), lateral parietal (LP), and medial prefrontal regions (MPFC) and considered the specificity of correlation in such regions with respect to the rest of the brain. A component was deemed a candidate DMN if it exhibited focal activity in at least two regions. In the rare instances where DMN activity appeared "split" between two hemispheric components (see results), the components were merged using the voxel-wise maximum operator before further evaluation.

SBA of the DMN was implemented by extracting two reference time-courses from the average of all voxels in the left and right PCC as defined below, and entering them as regressors in a first-level general-linear model analysis (Fox et al., 2005). For the purpose of the evaluations described below, the maps derived from the two hemispheric regressors were always combined using voxel-wise maximum operator, thresholded and considered together.

To obtain a further measure of intra- and inter-hemispheric connectivity across the DMN nodes, linear regressions were performed between the average BOLD signal time-courses in the left and right PCC, LP, and MPFC.

DEFAULT-MODE NETWORK EVALUATION

The detectability of the DMN was evaluated quantitatively with ICA as well as SBA for each preprocessing procedure P1–P5. In order to represent the intensity of DMN activity in the PCC, LP and MPFC, the peak z-score was calculated in the corresponding binary masks, obtained by intersecting the corresponding regions of the automated anatomical labeling atlas (AAL; Tzourio-Mazoyer et al., 2002) with the thresholded group-level DMN component maps from the controls, and dilated by 5 voxels to account for potential normalization imperfections. Due to their medial location, the left/right PCC and MPFC ROIs were contiguous; hence they were merged, yielding peak z-scores for bilateral PCC, bilateral MPFC, left LP (LP_L) and right LP (LP_R). To obtain a measure of correlation specificity, we additionally determined the extent of correlations outside the DMN regions by counting the voxels with a z-score > 2 and represented it as percentage of brain volume.

The presence of correlated activity in the DMN nodes on the ICA component map for the DMN was also visually rated by two experienced observers, blinded to participant information and preprocessing procedure, along the following scale: 0—definitely absent, 1—uncertain, 2—definitely present. The scores given by the two observers were averaged together; for patients the inter-rater agreement was 73%, 86%, 91%, and 88% for the PCC, MPFC, LP_L, and LP_R nodes. A global "DMN detectability" score was thereafter calculated summing the scores of the four nodes, and normalized within each patient with respect to the highest score attained individually; this step was introduced to reduce variability related to inter-participant differences, as the interest here was to compare the preprocessing procedures within each case.

STATISTICAL ANALYSIS

For all measures of interest a non-parametric related-samples Friedman test was performed, followed where appropriate by pair-wise Wilcoxon rank tests. Non-parametric tests were chosen in place of ANOVA since some distributions were significantly skewed. To account for multiple comparisons, all p-values were corrected using Bonferroni-Holm's procedure (Holm, 1979), performed over all Friedman and Wilcoxon tests, separately for patients and controls. To remove potential bias, scores corresponding to regions where the brain parenchyma was absent due to large anatomical lesions were removed and treated as missing values (10, 3, and 4 instances for the LP_R, LP_L, and MPFC, respectively) and imputed to the group median.

RESULTS

As indicated in **Table 1**, outlier volumes after rigid-body realignment were detected and rejected for 21 patients (70%) and 5 controls (15%). The results of statistical analyses are given in **Tables 2**, **3**.

In patients, for ICA assessed qualitatively elevating preprocessing grade increased between procedures P1 and P5 the number of patients for whom activity in each node was identifiable (**Table 4**); at group level, there was a significant difference between procedures P1–P2 only (**Figure 2**, **Table 3**). Elevating preprocessing grade also increased the peak correlation scores in the PCC and MPFC, with significant differences between procedures P1–P3 (**Figure 3B**, **Table 3**). Alongside improvement of correlation intensity, a reduction in the extent of spuriously correlated activity outside the DMN nodes was also observed, with significant difference between procedures P1–P3 (**Figure 4B**, **Table 3**). By contrast, in controls elevating preprocessing grade had no relevant effect on DMN detectability as assessed qualitatively (graph not shown). As reported in **Table 2**, elevating preprocessing grade nevertheless increased the peak correlation scores, but in this case the differences were primarily observed between procedures P3–P4 (**Figure 3A**); similar improvements were also detected for extra-DMN correlations (**Figure 4A**). While the levels of statistical significance of the effect of preprocessing were overall similar between patients and controls, it should be noted that the median

Table 1 | Statistics on the rejection of realignment-outlier volumes.

Any volume rejected?	Patients or controls	Number of subjects	Outliers detected	Volumes removed	Initial displacement (mm)		Residual displacement (mm)	
					Average	Worst	Average	Worst
No	Patients	9	0	0	0.06 ± 0.05	0.15 ± 0.13	0.06 ± 0.05	0.15 ± 0.13
Yes	Patients	21	20 ± 16	16 ± 23	0.13 ± 0.51	0.45 ± 1.85	0.12 ± 0.16	0.41 ± 0.44
Total	Patients	30	13 ± 16	10 ± 20	0.11 ± 0.35	0.45 ± 1.85	0.10 ± 0.12	0.41 ± 0.44
No	Controls	28	0	0	0.05 ± 0.04	0.11 ± 0.12	0.05 ± 0.04	0.11 ± 0.12
Yes	Controls	5	10 ± 7	0	0.08 ± 0.10	0.14 ± 0.17	0.08 ± 0.08	0.13 + 0.10
Total	Controls	33	1 ± 5	0	0.05 ± 0.05	0.14 ± 0.17	0.05 ± 0.05	0.13 + 0.10

In the subjects for whom outliers were detected, the residual displacement after rigid-body realignment of the time-series was reduced by outlier rejection. As described in-text, depending on the presence of contiguous outliers, replacement by interpolation or outright removal was performed. All values are given in mm as mean ± standard deviation of relative displacement between consecutive EPI volumes.

Table 2 | Statistical evaluation of the effect of the preprocessing procedures for healthy controls.

Analysis method	Parameter	Main effect (Friedman test)		*Post-hoc* (Wilcoxon signed ranks test)			
				P1–P2	P2–P3	P3–P4	P4–P5
		χ^2	*p*	*p*	*p*	*P*	*p*
Independent component analysis	Qualitative	8	1	–	–	–	–
	PCC: *z*-score	22	0.007	1	1	0.004	0.001
	MPFC: *z*-score	31	0.0001	0.3	0.03	0.005	0.1
	LP$_R$: *z*-score	17	0.07	–	–	–	–
	LP$_L$: *z*-score	23	0.006	1	0.6	0.0003	0.02
	Extra-DMN: %	39	<0.0001	0.6	0.02	0.004	0.1
Seed-based analysis	MPFC: *z*-score	71	<0.0001	0.003	0.0002	1	0.3
	LP$_R$: *z*-score	73	<0.0001	0.8	0.0005	1	1
	LP$_L$: *z*-score	65	<0.0001	1	0.0005	1	1
	Extra-DMN: %	120	<0.0001	<0.0001	<0.0001	0.0001	0.0001
Linear regression	LP$_L$-LP$_R$: r	100	<0.0001	0.002	<0.0001	0.03	0.03
	PCC$_L$-LP$_L$: r	112	<0.0001	<0.0001	<0.0001	0.2	0.6
	PCC$_R$-LP$_R$: r	104	<0.0001	0.0005	<0.0001	1	1
	PCC$_L$-MPFC$_L$: r	108	<0.0001	<0.0001	<0.0001	0.2	0.01
	PCC$_R$-MPFC$_R$: r	110	<0.0001	<0.0001	<0.0001	0.08	0.005

As described in-text, Friedman tests followed, where appropriate, by Wilcoxon post-hocs were performed. All p-values are reported following Bonferroni-Holm correction. PCC, posterior cingulate and precuneus; LP, lateral parietal cortex; MPFC, anterior cingulate and medial prefrontal cortex. Subscripts "L" or "R" stand for left or right hemisphere. "Extra-DMN" refers to the proportion of activated voxels outside the expected DMN localization. The effect of preprocessing "grade" was significant for most parameters, with the greatest overall differences being observed between P1–P2 and P2–P3, corresponding to the addition of band-pass filtering and removal of co-variance with the movement vectors and global brain parenchyma signal. See text for details.

magnitudes of the effect of preprocessing and inter-individual variability were substantially larger for patients (**Figures 3A** vs. **3B** and **4A** vs. **4B**).

For SBA, in patients elevating preprocessing grade markedly reduced the peak correlation scores across all regions, with a significant difference between procedures P2 and P3 for all regions (**Figure 3B**, **Table 3**). A strong reduction in the extent of spuriously correlated activity outside the DMN nodes was also apparent, with significant differences between all procedures (**Figure 4B**, **Table 3**). Similar effects were observed in controls (**Figures 3A**, **4A**).

For linear regression between average time-courses of the DMN nodes, preprocessing grade had a significant effect on all pairs investigated. In patients, the linear correlation coefficient between LP$_L$-LP$_R$, PCC$_L$-LP$_L$, and PCC$_R$-LP$_R$ monotonically decreased, whereas that between PCC$_L$-MPFC$_L$ and PCC$_R$-MPFC$_R$ displayed a more complex response, overall slightly increasing with disproportionately large values observed for procedure P2; here, significant *post-hoc* differences were found between procedures P1–P3 (**Figure 5B**, **Table 3**). The effect of preprocessing grade was more statistically significant in controls than patients primarily owing to substantially smaller

Table 3 | Statistical evaluation of the effect of the preprocessing procedures for patients.

Analysis method	Parameter	Main effect (Friedman test)		*Post-hoc* (Wilcoxon signed ranks test)			
				P1–P2	P2–P3	P3–P4	P4–P5
		χ^2	*p*	*p*	*p*	*P*	*p*
Independent component analysis	Qualitative	43	<0.0001	0.01	1	1	1
	PCC: *z*-score	31	0.0002	0.008	1	1	1
	MPFC: *z*-score	19	0.04	0.4	0.04	1	1
	LP$_R$: *z*-score	11	0.9	–	–	–	–
	LP$_L$: *z*-score	14	0.3	–	–	–	–
	Extra-DMN: %	48	<0.0001	0.002	0.02	1	1
Seed-based analysis	MPFC: *z*-score	49	<0.0001	1	0.002	1	0.8
	LP$_R$: *z*-score	58	< 0.0001	0.4	0.01	1	1
	LP$_L$: *z*-score	49	< 0.0001	0.4	0.006	1	0.9
	Extra-DMN: %	102	<0.0001	0.0002	0.0002	0.005	0.01
Linear regression	LP$_L$-LP$_R$: r	66	<0.0001	0.001	0.01	0.9	1
	PCC$_L$-LP$_L$: r	50	<0.0001	0.014	0.001	1	1
	PCC$_R$-LP$_R$: r	56	<0.0001	0.004	0.004	1	1
	PCC$_L$-MPFC$_L$: r	37	<0.0001	0.006	0.0008	1	0.1
	PCC$_R$-MPFC$_R$: r	37	<0.0001	0.003	0.0006	1	1

As described in-text, Friedman tests followed, where appropriate, by Wilcoxon post-hocs were performed. All p-values are reported following Bonferroni-Holm correction. PCC, posterior cingulate and precuneus; LP, lateral parietal cortex; MPFC, anterior cingulate and medial prefrontal cortex. Subscripts "L" or "R" stand for left or right hemisphere. "Extra-DMN" refers to the proportion of activated voxels outside the expected DMN localization. The effect of preprocessing "grade" was significant for most parameters, with the greatest overall differences being observed between P1–P2 and P2–P3, corresponding to the addition of band-pass filtering and removal of co-variance with the movement vectors and global brain parenchyma signal. See text for details.

Table 4 | Qualitative evaluation of default-mode network (DMN) node detectability on the component extracted by ICA.

Preprocessing procedure	Precuneus (PCC, 30 pts.) (%)	Right lateral partietal cortex (LP$_R$, 20 pts.) (%)	Left lateral parietal cortex (LP$_L$, 27 pts.) (%)	Medial prefrontal cortex (MPFC, 26 pts.) (%)
P1	14 (47)	12 (60)	9 (33)	8 (31)
P2	22 (73)	14 (70)	16 (59)	15 (58)
P3	23 (77)	16 (80)	17 (63)	14 (54)
P4	26 (87)	17 (85)	19 (70)	14 (54)
P5	27 (90)	16 (80)	20 (74)	14 (54)

The values represent the number and percentage of patients for whom activity was rated as definitely present. Percentages are adjusted to account for the number of cases where a node was affected by macroscopic anatomical damage (see Methods).

inter-individual variability. However, in absolute terms, the difference between P1 and P5 was larger in patients than controls for LP$_L$-LP$_R$, PCC$_L$-LP$_L$, and PCC$_R$-LP$_R$: while elevating preprocessing grade reduced median *r*-values down to about 0.7 in controls (**Figure 5A**), in patients removal of spurious signal reduced the median *r*-values to below 0.5, and for LP$_L$-LP$_R$ even below 0.2 (**Figure 5B**, **Table 2**). For PCC$_L$-MPFC$_L$ and PCC$_R$-MPFC$_R$, there was a converse pattern, wherein elevating preprocessing grade increased correlation much more for controls (up to about 0.6) than patients (about 0.3), plausibly representing the different effects of signal contamination on the anterior-posterior axis and poor preservation of MPFC connectivity in patients (**Figures 5A,B**, **Tables 2**, **3**).

Example ICA and SBA maps from representative cases are shown in **Figures 6–9**. Mirroring the numerical results reported in **Table 2**, ICA and SBA demonstrated a markedly different response to preprocessing grade. For ICA, improving filtering progressively enhanced the extent and intensity of correlation in the DMN nodes whereas for SBA, a gradual attenuation of diffuse, unspecific activity was observed. Reassuringly, as preprocessing was refined the results of ICA and SBA tended to converge. The entity of the effect of preprocessing at first could appear substantially larger for SBA than ICA, but one should consider that ICA failed to extract an identifiable DMN component in **Figures 7–9** unless procedure P3 or higher was utilized. Notably, in **Figure 7** an apparent "hemispheric splitting" of DMN activity is visible: the occurrence of this effect increased with preprocessing grade (i.e., P1: 2, P2: 5, P3: 6, P4: 9, and P5: 10 patients).

DISCUSSION

The present study extends previous comparative evaluations of rs-fMRI preprocessing (e.g., Murphy et al., 2009; Weissenbacher

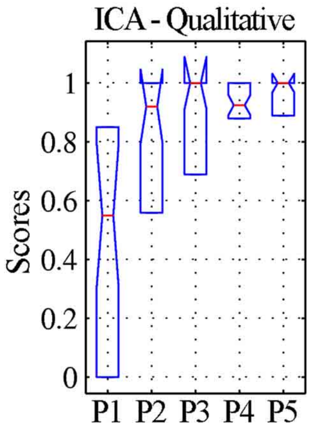


FIGURE 2 | Qualitative evaluation of the detectability of the whole default-mode network (DMN) extracted by ICA for patients. The values represent visual assessment scores for activity across the four main nodes (PCC, precuneus; LP_R, right lateral parietal cortex; LP_L, left lateral parietal cortex; MPFC, medial prefrontal cortex), averaged between the two raters and normalized within each patient, so that the maximum score of 1 corresponds to the best qualitative appearance of the DMN observed for each case (see text). The box-plots represent the median and inter-quartile ranges of the visual assessment scores. As preprocessing steps were added to the pipeline (P1–P5), the dispersion diminished and the median approached unity, confirming that the DMN was best identifiable after preprocessing using procedure P5.

et al., 2009; Van Dijk et al., 2010) to the specific population of patients in vegetative and minimally conscious state, which presents particular challenges related to substantial head movements and extensive anatomical damage. In this group, detection of residual neuroelectric activity is crucial for understanding disease staging and progression and the effect of possible rehabilitation therapies, and if rs-fMRI is performed to support clinical decision-making false negatives may have severe consequences; on the contrary, detection of DMN false-positives is less plausible. While deep sedation and even general anesthesia are routinely used for structural imaging in DoC patients, they would introduce severe confounds in rs-fMRI studies as they unavoidably depress central neural activity (e.g., Greicius et al., 2008; Boveroux et al., 2010; Deshpande et al., 2010; Stamatakis et al., 2010). Light sedation affects DMN activity more mildly but is generally insufficient to completely avoid movement (Giacino et al., 2006; Greicius et al., 2008).

As in Weissenbacher et al. (2009) the removal of co-variance with the movement vectors and average parenchyma signal coupled with BPF was found to have a substantial impact on the generation of DMN functional connectivity maps by both ICA and SBA. Here, additional steps to reject volumes contaminated by gross movement artifacts and to mask the ventricles were inserted and found to further improve data quality (**Tables 2–4**). The DMN maps generated with the two techniques showed a complementary sensitivity to preprocessing grade (i.e., procedures P1 to P5). For ICA, improving preprocessing resulted in larger and more significant correlations (**Figures 2**, **3**, **Tables 2**, **3**), with several cases in which an identifiable DMN component could not be extracted at all from data preprocessed with the most basic procedures (examples in **Figures 7–9**). By contrast, SBA maps initially displayed severe contamination by diffuse, unspecific correlations due to physiological noise and appeared progressively cleaner, with less significant but more focal and well-defined correlations for advanced preprocessing procedures (**Figures 3**, **4**). In line with previous reports, ICA was considerably less sensitive to the choice of preprocessing steps than SBA (Weissenbacher et al., 2009; Van Dijk et al., 2010; Power et al., 2012), especially in terms of separation of DMN activity from spurious correlations in other brain areas (**Figure 4** and examples in **Figures 6–9**) but it was clearly not indifferent. As indicated in **Tables 2**, **3**, ICA appeared relatively more sensitive to movement variance removal and BPF (P2), whereas SBA was most heavily influenced by removal of unspecific temporal variance (P3).

An important element of the proposed preprocessing chain is the automated removal of volumes contaminated by gross movement artifacts, which can be automatically identified as outliers on the basis of the residual root mean square intensity difference calculated during rigid-body realignment. This approach appears particularly convenient as it is completely operator-independent and straightforward to implement, with minimal assumptions on the type of artifacts Carp (2013). As indicated in **Figure 1**, it is important to reject any contaminated volumes prior to performing further preprocessing steps, namely slice-timing correction and temporal filtering, which entail assumptions on the relationships between consecutive time-points. The proposed "two-tier" approach, involving replacement by interpolation unless the number of contaminated volumes is excessive, has specific advantages in terms of minimizing the occurrence of undesirable temporal discontinuities (e.g., Carp, 2013). *Per-se*, the connectivity inferences drawn by both ICA and SBA are intrinsically insensitive to the removal and linear interpolation of time-points just as they are to temporal aliasing (Van Dijk et al., 2010).

In addition to removing spurious signal sources, the rejection of movement outliers also improves the proportion of real movement-related variance that can be removed through multilinear regression with respect to the movement vectors. This is a particularly important benefit, because in the presence of outliers the linear regression may be dominated by abnormally large or small signal levels for some volumes and thereby fail to properly capture the covariance with real movement. Movement can have substantial and highly region-dependent effects on the BOLD time-courses, as well-typified by the strong spectral correlations observed by Soddu et al. (2012) between the movement vectors and BOLD activity in a brain death patient.

Because DoC patients can present severely enlarged ventricles, due to increased intracranial pressure as well as atrophy, there is the possibility that the signal fluctuations due to pulsatile cerebrospinal fluid flow may be substantially over-represented with respect to a healthy brain, biasing the determination of the ICA un-mixing matrix (e.g., Power et al., 2012) and impairing the detection of weaker neuronal sources and affecting SBA through contamination of the seed signals by partial voluming with pulsating fluid in ventricles and sulci. Here, the effect of introducing the rejection of outlier volumes (procedure P4) and ventricle masking

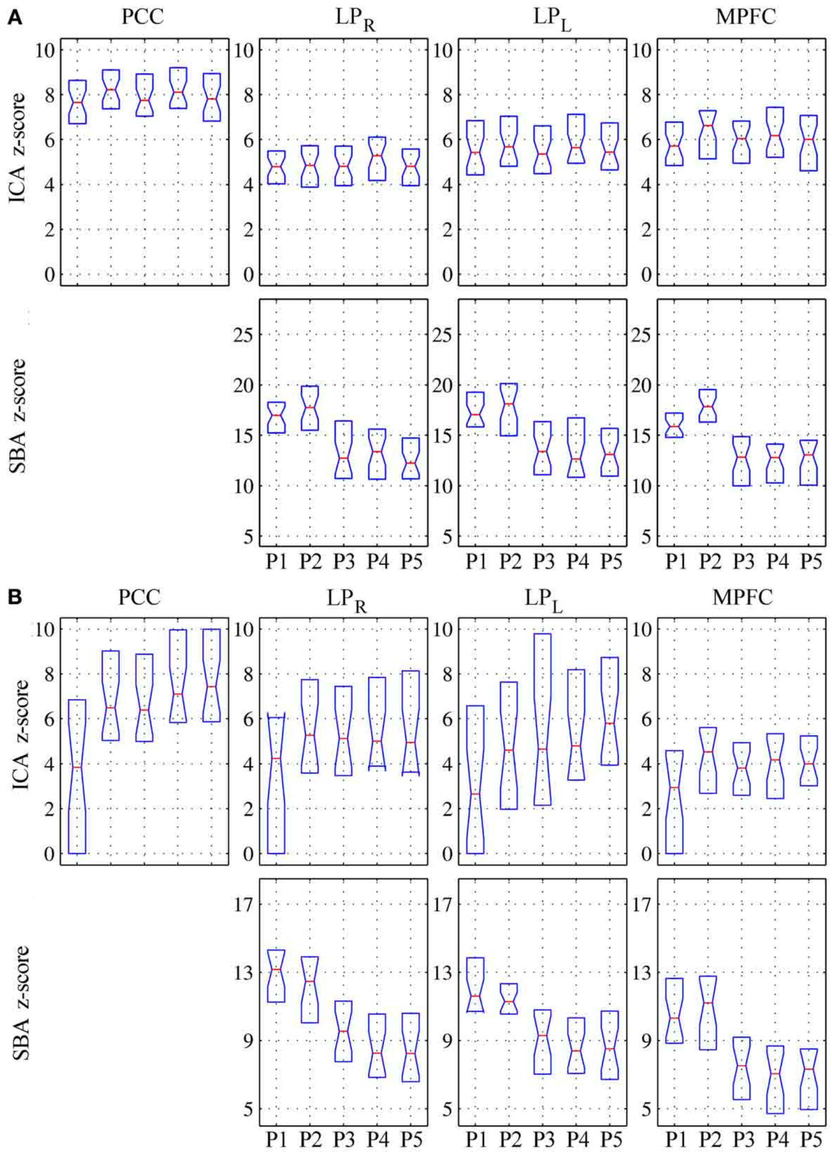


FIGURE 3 | Peak *z*-scores for activity within the four main DMN nodes for (A) healthy controls and (B) patients. Top row: DMN component extracted by ICA; bottom row: correlation maps computed using precuneus seeds (SBA). As preprocessing steps were added to the pipeline (P1–P5), the median *z*-scores for the DMN component extracted through ICA generally increased, indicating better component extraction, whereas the *z*-scores from SBA diminished (see text for comment and **Figures 6–9**).

(procedure P5) was more limited in comparison to that of BPF and removal of movement and global variance (procedures P2 and P3), yet at group level it was statistically significant for SBA, particularly reflecting into reduced extent of spurious correlations outside the expected DMN nodes. Importantly, even though for ICA the effect of these additional steps was not statistically significant at group level (**Table 3**), in several patients (e.g., **Table 4** and **Figures 7–9**) ICA failed to extract an identifiable DMN component for procedures P1–P3 but not P4–P5, and in specific cases (e.g., **Figure 6**) the visual appearance of activity in DMN nodes improved appreciably for procedure P5. Since procedures P4 and P5 are computationally parsimonious, we advise that they are always included in the preprocessing pipeline. While it may be argued that ICA should in principle not need any preprocessing thanks to its ability to isolate independent components, our data confirm that careful preprocessing is important not only for SBA, but also for ICA, as it improves component un-mixing and therefore reduces the risk of false negatives in DMN detection; of note, this effect was evident in patients but not in controls, plausibly reflecting differences in the entity of movement artifacts and strength of component signals.

Further insight into the effect of movement on correlations between regional time-courses is provided by the linear regression analyses. As discussed in Power et al. (2012) and Satterthwaite et al. (2013), correlation between two regions can be inflated if they undergo a common translation or masked if they undergo a rotation around a point located between them. In other words, head movement artifactually increases functional coupling across local networks and decreases it for long-range connections (Power et al., 2012; Van Dijk et al., 2012). Here, elevating preprocessing grade had markedly different effects on the evaluation of the shorter latero-lateral connections between the lateral parietal cortex and the precuneus and longer anterior-posterior connections between the precuneus and the medial prefrontal cortex (**Figure 5**). In the first case, a gradual reduction of the correlation coefficients was observed, signaling that a substantial part of the raw covariance was induced by movement and global fluctuations. In the second case, a biphasic response emerged: while the correlation coefficients overall increased, initially regressing-out movement-related variance (procedure P2) boosted the correlations but subsequently eliminating global variance reduced them again (procedure P3). This suggests that movement initially masked the covariance between these regions, which was, however, dominated by unspecific, global fluctuations in patients. In controls, the effect of preprocessing procedure on latero-lateral connections was more constrained, plausibly due to less movement, but greater changes were observed in anterior-posterior connectivity with respect to patients: this plausibly reflects the fact that frontal DMN connectivity is strong in controls but very weak or lost in patients, and may also be related to different movement patterns along the three axes. These different effects along the lateral and anterior-posterior directions agree with previous investigations and further underline the potential for complex confounding effects (Power et al., 2012; Van Dijk et al., 2012; Satterthwaite et al., 2013), stressing the importance of adopting comprehensive preprocessing approaches that attempt to eliminate as much spurious signal variance as possible.

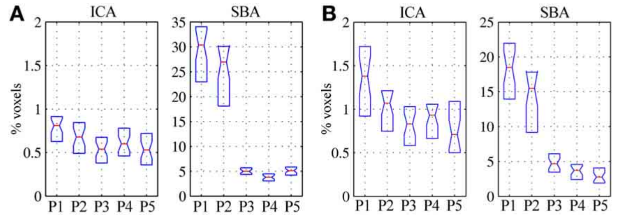


FIGURE 4 | Volume of significant activations (z = 2) outside the regions-of-interest covering the expected DMN nodes (i.e., PCC, LP$_{\text{R}}$, LP$_{\text{L}}$, and MPFC), for the DMN maps extracted through ICA (left) and SBA (right), expressed as percent with respect to the parenchymal volume for (A) healthy controls and (B) patients. As preprocessing steps were added to the pipeline (P1–P5), the extent of activations outside the expected localization of the DMN was progressively reduced, representing greater specificity of the functional connectivity maps; the effect was considerably more marked for SBA than ICA.

The present study has several limitations that need to be considered. First, because no gold-standard reference for DMN activity is available, the evaluation necessarily remains an empirical one, and it is not possible to formally confirm how much of the signal variance eliminated in each step was artifactual rather than neural. Yet, since the DMN has a highly stereotyped appearance (i.e., is expected to involve specific nodes at relatively stable anatomical locations), its increased detectability reassured on the overall beneficial effect of the suggested preprocessing techniques (Esposito et al., 2008). Second, almost all DoC patients are characterized by extensive brain anatomical abnormalities, and the present study did not consider in detail the effect of imperfect normalization caused by poor structural similarity between the damaged individual brains and the standardized healthy brain template. This issue, which equally affects all other studies in

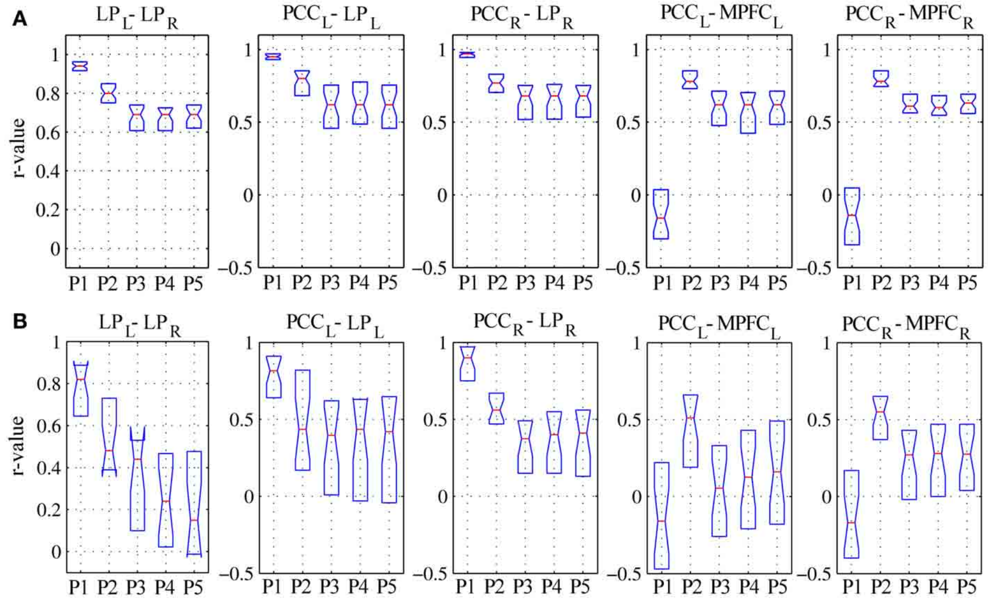


FIGURE 5 | Linear correlation coefficients for regionally-averaged BOLD signal time-series between DMN regions for (A) healthy controls and (B) patients. See text for description of the results.

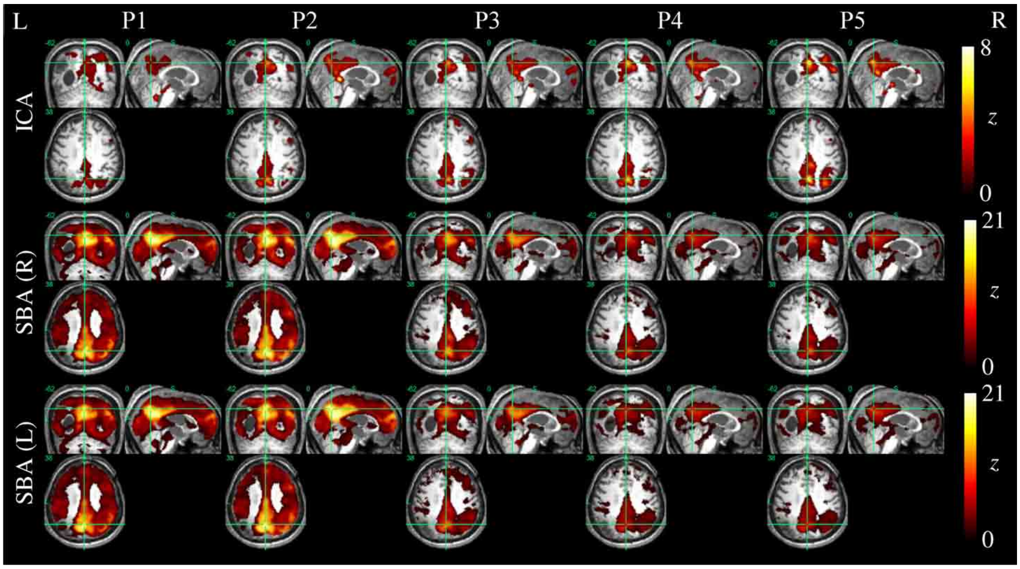


FIGURE 6 | DMN functional connectivity maps computed with ICA (top row) and SBA (middle and bottom rows) for a patient with a clinical diagnosis of vegetative state (maximum displacement 4.0 mm, 27/200 outlier volumes). As preprocessing steps were added (left to right, P1–P5), activity in the right angular gyrus became more evident on the ICA maps. For SBA, enhanced preprocessing had the effect of progressively reducing the diffuse correlations observed throughout the brain, revealing a topographical pattern that converged to that extracted by ICA.

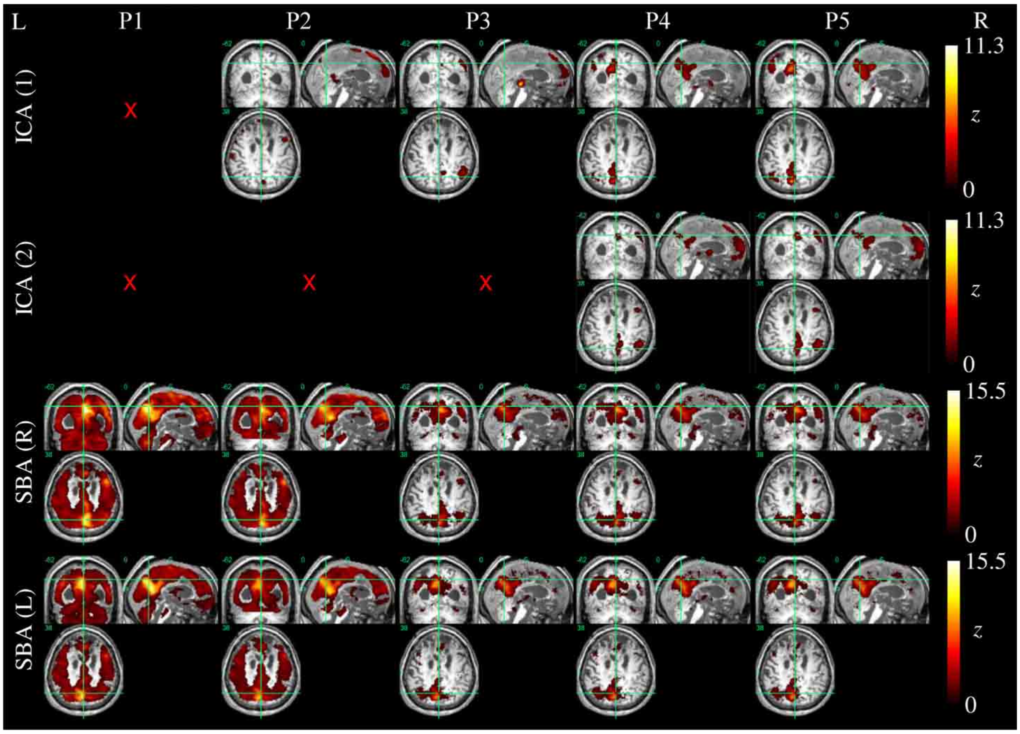


FIGURE 7 | DMN functional connectivity maps computed with ICA (top row) and SBA (middle and bottom rows) for a patient with a clinical diagnosis of vegetative state (maximum displacement 0.8 mm, 3/200 outlier volumes); red crosses denote inability to identify DMN activity in any of the 20 components extracted by ICA. As preprocessing grade was elevated (left to right, P1–P5), coherent activity between the precuneus and the angular gyri became identifiable through ICA and SBA. Notably, in this patient an apparent "split" between left and right DMN connectivity was observed through both analyses.

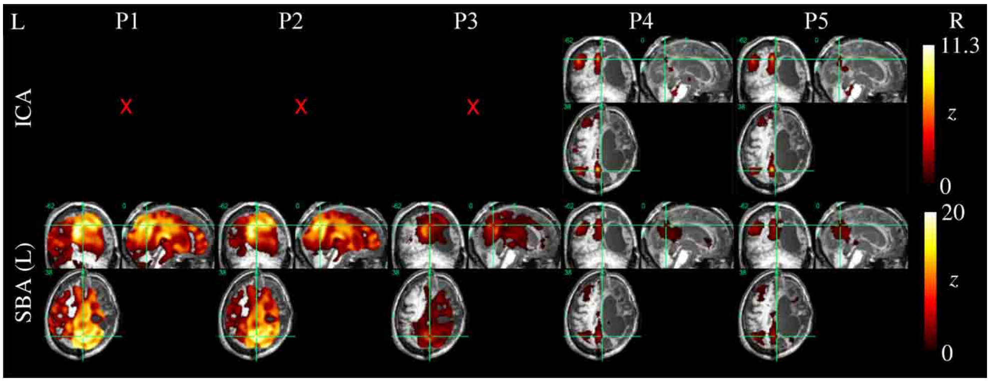


FIGURE 8 | DMN functional connectivity maps computed with ICA (top row) and SBA (middle and bottom rows) for a patient with a clinical diagnosis of minimally-conscious state (maximum displacement 20.3 mm, 16/200 outlier volumes); red crosses denote inability to identify DMN activity in any of the 20 components extracted by ICA. Due to the gross anatomical damage visible on the volumetric T_1 scan, SBA with the right precuneus seed was not performed. Here, elevating preprocessing grade (left to right, P1–P5) revealed coherent activity between the left precuneus and angular gyrus: applying procedures 4 and 5, ICA decomposition became able to orthogonalize activity for this preserved DMN subset, and SBA maps were "cleaned" of unspecific physiological fluctuations that originally extended to areas of gross anatomical damage.

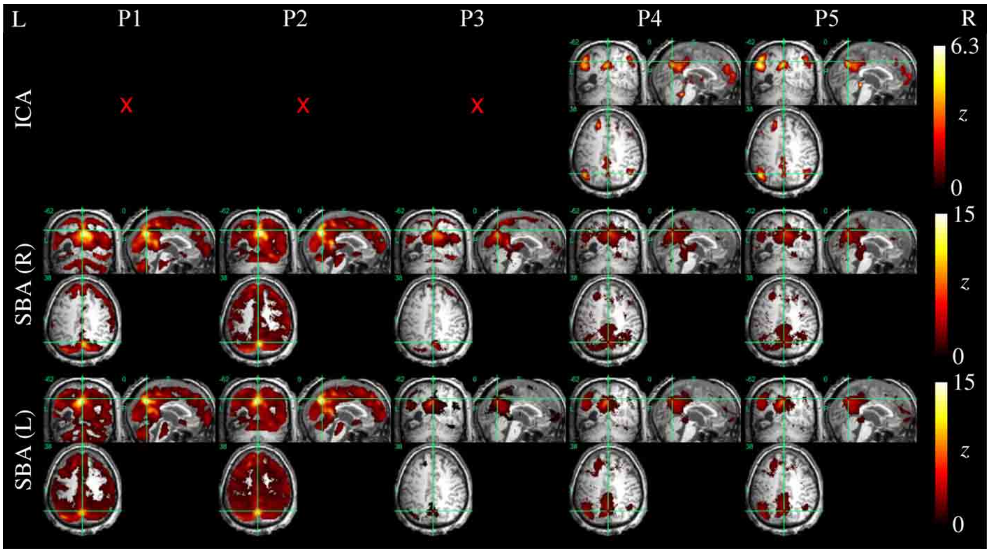


FIGURE 9 | DMN functional connectivity maps computed with ICA (top row) and SBA (middle and bottom rows) for a patient with a clinical diagnosis of minimally-conscious state (maximum movement 5.0 mm, 24/200 outlier volumes); red crosses denote inability to identify DMN activity in any of the 20 components extracted by ICA. Here, applying procedures 4 and 5 made ICA decomposition capable of revealing coherent activity between the precuneus, angular gyri and a cluster in the left superior frontal lobe. For SBA, a non-monotonic effect is evident, whereby applying procedure 3 removed substantial unspecific covariation across the brain parenchyma, and the subsequent steps implemented in procedures 4 and 5 revealed coherent activity between the precuneus, angular gyrus and superior frontal lobe.

this area, will need to be evaluated in future studies. Third, anti-correlations were not considered and the evaluation of component detectability with ICA and SBA was limited, as in some other studies in this area, to positive correlations. While there is increasing evidence that negative correlations may also represent architecturally important forms of functional connectivity, the interpretation of such effects remains unclear, hence they were not considered here (e.g., Rosazza and Minati, 2011). Fourth, we did not include relative displacements in our regressors as advised by Satterthwaite et al. (2013) and Power et al. (2012); this parameter needs to be considered in future work. Our approach otherwise maps closely with their suggestions and includes additional steps that are specifically beneficial in this population where movement is substantial and requires outright rejection of volumes and the extent of atrophy and weak signal justify the masking of non-brain structures. Fifth, the DMN component was selected and rated manually by expert operators. Spatial templates of the DMN (Esposito et al., 2008) are available, as well as automated techniques based on multi-dimensional "fingerprints" and support vector machines which have been successfully applied to data from DoC patients (Soddu et al., 2012), and the effect of preprocessing choices on their performance should be evaluated in future work (De Martino et al., 2007). An inherent issue is the inability to quantify the proportion of false negatives: because no gold-standard of DMN integrity exists, our data suggest that elevating preprocessing grade reduces the incidence of false negatives, but do not enable quantifying the risk of false negatives. That parameter will need to be determined in future studies addressing correlation with clinical status as well as test-retest reliability. Finally, recent work has demonstrated the possibility to explicitly extract the cardiac and respiratory regressors directly from the fMRI data (Beall, 2010), and additionally offered specific advice regarding movement regressor filtering (Hallquist et al., 2013) and rejection of movement-contaminated data (Christodoulou et al., 2013); the corresponding techniques will need to be considered to update and extend the results obtained in the present investigation.

CONCLUSION

This study provides a comprehensive evaluation of the effect of data preprocessing choices on rs-fMRI in DoC patients, and corroborates the existing literature in this area through systematic comparison of five preprocessing procedures of increasing complexity. ICA and SBA were both found to be significantly impacted by data preprocessing settings, albeit with different patterns. Elevating preprocessing grade improved the ability of ICA to successfully un-mix DMN activity and generally enhanced the significance and extent of DMN correlations. By contrast, for SBA high-grade preprocessing had the principal effect of reducing contamination by unspecific, systemic signal sources, reflecting in progressively more focal and well-defined activity patterns. As preprocessing grade was elevated, the topographical maps provided by the two techniques tended to converge. The results strongly underline the importance of performing high-grade preprocessing, including rejection of outlier volumes, ventricle masking, removal of movement related and global signal covariance and BPF. Even though a gold-standard measure of connectivity preservation does not exist, since the DMN has highly characteristic topographical features the observation that its detectability increases with better preprocessing indicates reduced risk of false negative errors. We propose that the described preprocessing procedures should be adopted as a minimum standard in future studied in this area to reduce the probability of wrongly inferring that DMN activity is absent, with potential implications for clinical management.

ACKNOWLEDGMENTS

The start-up Coma Research Centre (CRC) project was funded by a healthcare grant from the Lombardia regional government. The authors are grateful to Francesca Epifani, Anna Nigri, Nicola Bertolino, Stefania Ferraro and Francesco Ghielmetti for excellent support during data acquisition and insightful advice. The Coma Research Centre (CRC) multidisciplinary team, on behalf of which the present publication was submitted, acknowledges the following members: Andronache A, Benti R, Bruzzone MG, Caldiroli D, Ciaraffa F, Covelli V, D'Incerti L, Fazio P, Ferraro S, Franceschetti S, Giovannetti A, Leonardi M, Marotta G, Minati L, Molteni F, Nigri A, Pagani M, Panzica F, Parati EA, Reggiori B, Rosazza C, Rossi D, Sattin D, Varotto G, Vela Gomez J.

REFERENCES

Allen, E. A., Erhardt, E. B., Damaraju, E., Gruner, W., Segall, J. M., Silva, R. F., et al. (2011). A baseline for the multivariate comparison of resting-state networks. *Front. Syst. Neurosci.* 5:2. doi: 10.3389/fnsys.2011.00002

Beall, E. B. (2010). Adaptive cyclic physiologic noise modeling and correction in functional MRI. *J. Neurosci. Methods* 187, 216–228. doi: 10.1016/j.jneumeth.2010.01.013

Biswal, B. B., Mennes, M., Zuo, X. N., Gohel, S., Kelly, C., Smith, S. M., et al. (2010). Toward discovery science of human brain function. *Proc. Natl. Acad. Sci. U.S.A.* 107, 4734–4739. doi: 10.1073/pnas.0911855107

Birn, R. M., Diamond, J. B., Smith, M. A., and Bandettini, P. A. (2006). Separating respiratory-variation-related fluctuations from neuronal-activity-related fluctuations in fMRI. *Neuroimage* 31, 1536–1548. doi: 10.1016/j.neuroimage.2006.02.048

Boly, M., Tshibanda, L., Vanhaudenhuyse, A., Noirhomme, Q., Schnakers, C., Ledoux, D., et al. (2009). Functional connectivity in the default network during resting state is preserved in a vegetative but not in a brain dead patient. *Hum. Brain Mapp.* 30, 2393–2400. doi: 10.1002/hbm.20672

Boveroux, P., Vanhaudenhuyse, A., Bruno, M. A., Noirhomme, Q., Lauwick, S., Luxen, A., et al. (2010). Breakdown of within- and between-network resting state functional magnetic resonance imaging connectivity during propofol-induced loss of consciousness. *Anesthesiology* 113, 1038–1053. doi: 10.1097/ALN.0b013e3181f697f5

Buckner, R. L., Andrews-Hanna, J. R., and Schacter, D. L. (2008). The brain's default network: anatomy, function, and relevance to disease. *Ann. N.Y. Acad. Sci.* 1124, 1–38. doi: 10.1196/annals.1440.011

Calhoun, V. D., Adali, T., Pearlson, G. D., and Pekar, J. J. (2001). A method for making group inferences from functional MRI data using independent component analysis. *Hum. Brain Mapp.* 14, 140–151. doi: 10.1002/hbm.1048

Calhoun, V. D., Kiehl, K. A., and Pearlson, G. D. (2008). Modulation of temporally coherent brain networks estimated using at rest and during cognitive tasks. *Hum. Brain Mapp.* 29, 828–838. doi: 10.1002/hbm.20581

Carp, J. (2013). Optimizing the order of operations for movement scrubbing: comment on Power et al. *Neuroimage* 76, 436–438. doi: 10.1016/j.neuroimage.2011.12.061

Cauda, F., Micon, B. M., Sacco, K., Duca, S., D'Agata, F., Geminiani, G., et al. (2009). Disrupted intrinsic functional connectivity in the vegetative state. *J. Neurol. Neurosurg. Psychiatry* 80, 429–431. doi: 10.1136/jnnp.2007.142349

Christodoulou, A. G., Bauer, T. E., Kiehl, K. A., Feldstein Ewing, S. W., Bryan, A. D., and Calhoun, V. D. (2013). A quality control method for detecting and suppressing uncorrected residual motion in fMRI studies. *Magn. Reson. Imaging* 31, 707–717. doi: 10.1016/j.mri.2012.11.007

Corfield, D. R., Murphy, K., Josephs, O., Adams, L., and Turner, R. (2001). Does hypercapnia-induced cerebral vasodilation modulate the hemodynamic response to neural activation. *Neuroimage* 13, 1207–1211. doi: 10.1006/nimg.2001.0760

De Martino, F., Gentile, F., Esposito, F., Balsi, M., Di Salle, F., Goebel, R., et al. (2007). Classification of fMRI independent components using IC-fingerprints and support vector machine classifiers. *Neuroimage* 34, 177–194. doi: 10.1016/j.neuroimage.2006.08.041

Deshpande, G., Kerssens, C., Sebel, P. S., and Hu, X. (2010). Altered local coherence in the default mode network due to sevoflurane anesthesia. *Brain Res.* 1318, 110–121. doi: 10.1016/j.brainres.2009.12.075

Desjardins, A. E., Kiehl, K. A., and Liddle, P. F. (2001). Removal of confounding effects of global signal in functional MRI analyses. *Neuroimage* 13, 751–758.

Esposito, F., Aragri, A., Pesaresi, I., Cirillo, S., Tedeschi, G., Marciano, E., et al. (2008). Independent component model of the default-mode brain function: combining individual-level and population-level analyses in resting-state fMRI. *Magn. Reson. Imaging* 26, 905–913. doi: 10.1016/j.mri.2008.01.045

Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S., and Turner, R. (1996). Movement-related effects in fMRI time-series. *Magn.*

Reson. Med. 35, 346–355. doi: 10.1002/mrm.1910350312

Fox, M. D., and Raichle, M. E. (2007). Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nat. Rev. Neurosci.* 8, 700–711. doi: 10.1038/nrn2201

Fox, M. D., Snyder, A. Z., Vincent, J. L., Corbetta, M., Van Essen, D. C., and Raichle, M. E. (2005). The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proc. Natl. Acad. Sci. U.S.A.* 102, 9673–9678. doi: 10.1073/pnas.0504136102

Fox, M. D., Zhang, D., and Raichle, M. E. (2009). The global signal and observed anticorrelated resting state brain networks. *J. Neurophysiol.* 101, 3270–3283. doi: 10.1152/jn.90777.2008

Giacino, J. T. (2004). The vegetative and minimally conscious states: consensus-based criteria for establishing diagnosis and prognosis. *NeuroRehabilitation* 19, 293–298.

Giacino, J. T., Hirsch, J., Schiff, N., and Laureys, S. (2006). Functional neuroimaging applications for assessment and rehabilitation planning in patients with disorders of consciousness. *Arch. Phys. Med. Rehabil.* 87, S67–S76. doi: 10.1016/j.apmr.2006.07.272

Giacino, J. T., Kalmar, K., and Whyte, J. (2004). The JFK coma recovery scale-revised: measurement characteristics and diagnostic utility. *Arch. Phys. Med. Rehabil.* 85, 2020–2029. doi: 10.1016/j.apmr.2004.02.033

Greicius, M. D., Kiviniemi, V., Tervonen, O., Vainionpää, V., Alahuhta, S., Reiss, A. L., et al. (2008). Persistent default-mode network connectivity during light sedation. *Hum. Brain Mapp.* 29, 839–847. doi: 10.1002/hbm.20537

Hallquist, M. N., Hwang, K., and Luna, B. (2013). The nuisance of nuisance regression: spectral misspecification in a common approach to resting-state fMRI preprocessing reintroduces noise and obscures functional connectivity. *Neuroimage* 82C, 208–225. doi: 10.1016/j.neuroimage.2013.05.116

Hutton, C., Bork, A., Josephs, O., Deichmann, R., Ashburner, J., and Turner, R. (2002). Image distortion correction in fMRI: a quantitative evaluation. *Neuroimage* 16, 217–240. doi: 10.1006/nimg.2001.1054

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* 6, 65–70.

Johnstone, T., Ores Walsh, K. S., Greischar, L. L., Alexander, A. L., Fox, A. S., Davidson, R. J., et al. (2006). Motion correction and the use of motion covariates in multiple-subject fMRI analysis. *Hum. Brain Mapp.* 27, 779–788. doi: 10.1002/hbm.20219

Laureys, S. (2005). The neural correlate of (un)awareness: lessons from the vegetative state. *Trends Cogn. Sci.* 9, 556–559. doi: 10.1016/j.tics.2005.10.010

Lombardi, F., Gatta, G., Sacco, S., Muratori, A., and Carolei, A. (2007). The Italian version of the coma recovery scale-revised (CRS-R). *Funct. Neurol.* 22, 47–61.

Lund, T. E., Madsen, K. H., Sidaros, K., Luo, W. L., and Nichols, T. E. (2006). Non-white noise in fMRI: does modelling have an impact. *Neuroimage* 29, 54–66. doi: 10.1016/j.neuroimage.2005.07.005

Macey, P. M., Macey, K. E., Kumar, R., and Harper, R. M. (2004). A method for removal of global effects from fMRI time series. *Neuroimage* 22, 360–366. doi: 10.1016/j.neuroimage.2003.12.042

Multi-Society Task Force Report on PVS. (1994). Medical aspects of the persistent vegetative state. *N. Engl. J. Med.* 330, 1499–508. doi: 10.1056/NEJM199405263302107

Murphy, K., Birn, R. M., Handwerker, D. A., Jones, T. B., and Bandettini, P. A. (2009). The impact of global signal regression on resting state correlations: are anti-correlated networks introduced. *Neuroimage* 44, 893–905. doi: 10.1016/j.neuroimage.2008.09.036

Noirhomme, Q., Soddu, A., Lehembre, R., Vanhaudenhuyse, A., Boveroux, P., Boly, M., et al. (2010). Brain connectivity in pathological and pharmacological coma. *Front. Syst. Neurosci.* 4:160. doi: 10.3389/fnsys.2010.00160

Owen, A. M., and Coleman, M. R. (2007). Functional MRI in disorders of consciousness: advantages and limitations. *Curr. Opin. Neurol.* 20, 632–637. doi: 10.1097/WCO.0b013e3282f15669

Owen, A. M., and Coleman, M. R. (2008). Functional neuroimaging of the vegetative state. *Nat. Rev. Neurosci.* 9, 235–243. doi: 10.1038/nrn2330

Owen, A. M., Coleman, M. R., Davis, M. H., Boly, M., Laureys, S., and Pickard, J. D. (2006). Detecting awareness in the vegetative state. *Science* 313, 1402. doi: 10.1126/science.1130197

Power, J. D., Barnes, K. A., Schlaggar, B. L., and Petersen, S. E. (2012). Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. *Neuroimage* 59, 2142–2154. doi: 10.1016/j.neuroimage.2011.10.018

Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001). A default mode of brain function. *Proc. Natl. Acad. Sci. U.S.A.* 98, 676–682. doi: 10.1073/pnas.98.2.676

Rappaport, M. (2005). The disability rating and coma/near-coma scales in evaluating severe head injury. *Neuropsychol. Rehabil.* 15, 442–453. doi: 10.1080/09602010443000335

Rosazza, C., and Minati, L. (2011). Resting-state brain networks: literature review and clinical applications. *Neurol. Sci.* 32, 773–785. doi: 10.1007/s10072-011-0636-y

Rosazza, C., Minati, L., Ghielmetti, F., Mandelli, M. L., and Bruzzone, M. G. (2012). Functional connectivity during resting-state functional MR imaging: study of the correspondence between independent component analysis and region-of-interest-based methods. *AJNR Am. J. Neuroradiol.* 33, 180–187. doi: 10.3174/ajnr.A2733

Satterthwaite, T. D., Elliott, M. A., Gerraty, R. T., Ruparel, K., Loughead, J., Calkins, M. E., et al. (2013). An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data. *Neuroimage* 64, 240–256. doi: 10.1016/j.neuroimage.2012.08.052

Stamatakis, E. A., Adapa, R. M., Absalom, A. R., and Menon, D. K. (2010). Changes in resting neural connectivity during propofol sedation. *PLoS ONE* 5:e14224. doi: 10.1371/journal.pone.0014224

Soddu, A., Vanhaudenhuyse, A., Bahri, M. A., Bruno, M. A., Boly, M., Demertzi, A., et al. (2012). Identifying the default-mode component in spatial IC analyses of patients with disorders of consciousness. *Hum. Brain Mapp.* 33, 778–796. doi: 10.1002/hbm.21249

Soddu, A., Vanhaudenhuyse, A., Demertzi, A., Bruno, M. A., Tshibanda, L., Di, H., et al. (2011). Resting state activity in patients with disorders of consciousness. *Funct. Neurol.* 26, 37–43.

Strother, S. C. (2006). Evaluating fMRI preprocessing pipelines. *IEEE Eng. Med. Biol.* 25, 27–41. doi: 10.1109/MEMB.2006.1607667

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., et al. (2002). Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage* 15, 273–289. doi: 10.1006/nimg.2001.0978

Van Dijk, K. R., Hedden, T., Venkataraman, A., Evans, K. C., Lazar, S. W., and Buckner, R. L. (2010). Intrinsic functional connectivity as a tool for human connectomics: theory, properties, and optimization. *J. Neurophysiol.* 103, 297–321. doi: 10.1152/jn.00783.2009

Van Dijk, K. R., Sabuncu, M. R., and Buckner, R. L. (2012). The influence of head motion on intrinsic functional connectivity MRI. *Neuroimage* 59, 431–438. doi: 10.1016/j.neuroimage.2011.07.044

Vanhaudenhuyse, A., Noirhomme, Q., Tshibanda, L. J., Bruno, M. A., Boveroux, P., Schnakers, C., et al. (2010). Default network connectivity reflects the level of consciousness in non-communicative brain-damaged patients. *Brain* 133, 161–171. doi: 10.1093/brain/awp313

Weissenbacher, A., Kasess, C., Gerstl, F., Lanzenberger, R., Moser, E., and Windischberger, C. (2009). Correlations and anticorrelations in resting-state functional connectivity MRI: a quantitative comparison of preprocessing strategies. *Neuroimage* 47, 1408–1416. doi: 10.1016/j.neuroimage.2009.05.005

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 04 July 2013; accepted: 30 July 2013; published online: 22 August 2013.

Citation: Andronache A, Rosazza C, Sattin D, Leonardi M, D'Incerti L and Minati L (2013) Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness. Front. Neuroinform. 7*:16. doi: 10.3389/fninf.2013.00016*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 08 April 2014
doi: 10.3389/fninf.2014.00031

Machine learning patterns for neuroimaging-genetic studies in the cloud

Benoit Da Mota[1,2]*, Radu Tudoran[3], Alexandru Costan[3], Gaël Varoquaux[1,2], Goetz Brasche[4], Patricia Conrod[5,6], Herve Lemaitre[7], Tomas Paus[8,9,10], Marcella Rietschel[11,12], Vincent Frouin[2], Jean-Baptiste Poline[2,13], Gabriel Antoniu[3], Bertrand Thirion[1,2]* and IMAGEN Consortium[14]

[1] Parietal Team, INRIA Saclay, Île-de-France, Saclay, France
[2] CEA, DSV, I²BM, Neurospin, Gif-sur-Yvette, France
[3] KerData Team, INRIA Rennes - Bretagne Atlantique, Rennes, France
[4] Microsoft, Advance Technology Lab Europe, Munich, Germany
[5] Institute of Psychiatry, King's College London, London, UK
[6] Department of Psychiatry, Universite de Montreal, CHU Ste Justine Hospital, Montreal, QC, Canada
[7] Institut National de la Santé et de la Recherche Médicale, INSERM CEA Unit 1000 "Imaging & Psychiatry," University Paris Sud, Orsay, and AP-HP Department of Adolescent Psychopathology and Medicine, Maison de Solenn, University Paris Descartes, Paris, France
[8] Rotman Research Institute, University of Toronto, Toronto, ON, Canada
[9] School of Psychology, University of Nottingham, Nottingham, UK
[10] Montreal Neurological Institute, McGill University, Montréal, QC, Canada
[11] Central Institute of Mental Health, Mannheim, Germany
[12] Medical Faculty Mannheim, University of Heidelberg, Heidelberg, Germany
[13] Henry H. Wheeler Jr. Brain Imaging Center, University of California at Berkeley, Berkeley, CA, USA
[14] www.imagen-europe.com

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Mensur Dlakic, Montana State University, USA
Roman Moucek, University of West Bohemia, Czech Republic

****Correspondence:***
Benoit Da Mota and Bertrand Thirion, Parietal Team, INRIA Saclay, Île-de-France, Neurospin, CEA Saclay, Bat 145, PC 156, 91191 Gif sur Yvette, Saclay, France
e-mail: benoit.da_mota@inria.fr; bertrand.thirion@inria.fr

Brain imaging is a natural intermediate phenotype to understand the link between genetic information and behavior or brain pathologies risk factors. Massive efforts have been made in the last few years to acquire high-dimensional neuroimaging and genetic data on large cohorts of subjects. The statistical analysis of such data is carried out with increasingly sophisticated techniques and represents a great computational challenge. Fortunately, increasing computational power in distributed architectures can be harnessed, if new neuroinformatics infrastructures are designed and training to use these new tools is provided. Combining a MapReduce framework (TomusBLOB) with machine learning algorithms (Scikit-learn library), we design a scalable analysis tool that can deal with non-parametric statistics on high-dimensional data. End-users describe the statistical procedure to perform and can then test the model on their own computers before running the very same code in the cloud at a larger scale. We illustrate the potential of our approach on real data with an experiment showing how the functional signal in subcortical brain regions can be significantly fit with genome-wide genotypes. This experiment demonstrates the scalability and the reliability of our framework in the cloud with a 2 weeks deployment on hundreds of virtual machines.

Keywords: machine learning, neuroimaging-genetic, cloud computing, fMRI, heritability

1. INTRODUCTION

Using genetics information in conjunction with brain imaging data is expected to significantly improve our understanding of both normal and pathological variability of brain organization. It should lead to the development of biomarkers and in the future personalized medicine. Among other important steps, this endeavor requires the development of adapted statistical methods to detect significant associations between the highly heterogeneous variables provided by genotyping and brain imaging, and the development of software components with which large-scale computation can be done.

In current settings, neuroimaging-genetic datasets consist of a set of (1) genotyping measurements at given genetic loci, such as Single Nucleotide Polymorphisms (SNPs) that represent a large amount of the genetic between-subject variability, and (2) quantitative measurements at given locations (voxels) in three-dimensional images, that represent e.g., either the amount of functional activation in response to a certain task or an anatomical feature, such as the density of gray matter in the corresponding brain region. These two sets of features are expected to reflect differences in brain organization that are related to genetic differences across individuals.

Most of the research efforts so far have been focused on designing association models, while the computational procedures used to run these models on actual architectures have not been considered carefully. Voxel intensity and cluster size methods have been used for genome-wide association studies (GWAS) (Stein et al., 2010), but the multiple comparisons problem most often does not permit to find significant results, despite efforts to estimate the effective number of tests (Gao et al., 2010) or by paying the cost of a permutation test (Da Mota et al., 2012). Working at the genes level instead of SNPs (Hibar et al., 2011; Ge et al., 2012) is a promising approach, especially if we are looking at monogenic (or few causal genes) diseases.

For polygenic diseases, gains in sensitivity might be provided by multivariate models in which the joint variability of several genetic variables is considered simultaneously. Such models are thought to be more powerful (Meinshausen and Bühlmann, 2010; Vounou et al., 2010; Bunea et al., 2011; Kohannim et al., 2011; Floch et al., 2012), because they can express more complex relationships than simple pairwise association models. The cost of unitary fit is high due to high-dimensional, potentially non-smooth optimization problems and various cross-validation loops needed to optimize the parameters; moreover, permutation testing is necessary to assess the statistical significance of the results of such procedures in the absence of analytical tests. Multivariate statistical methods require thus many efforts to be tractable for this problem on both the algorithmic and implementation side, including the design of adapted dimension reduction schemes. Working in a distributed context is necessary to deal efficiently with the memory and computational loads.

Today, researchers have access to many computing capabilities to perform data-intensive analysis. The cloud is increasingly used to run such scientific applications, as it offers a reliable, flexible, and easy to use processing pool (Vaquero et al., 2008; Jackson et al., 2010; Hiden et al., 2012; Juve et al., 2012). The MapReduce paradigm (Chu et al., 2006; Dean and Ghemawat, 2008) is the natural candidate for these applications, as it can easily scale the computation by applying in parallel an operation on the input data (map) and then combine these partials results (reduce). However, some substantial challenges still have to be addressed to fully exploit the power of cloud infrastructures, such as data access, as it is currently achieved through high latency protocols, which are used to access the cloud storage services (e.g., Windows Azure Blob). To sustain geographically distributed computation, the storage system needs to manage concurrency, data placement and inter-site data transfers.

We propose an efficient framework that can manage inferences on neuroimaging-genetic studies with several phenotypes and permutations. It combines a MapReduce framework (TomusBLOB, Costan et al., 2013) with machine learning algorithms (Scikit-learn library) to deliver a scalable analysis tool. The key idea is to provide end-users the capability to easily describe the statistical inference that they want to perform and then to test the model on their own computers before running the very same code in the cloud at a larger scale. We illustrate the potential of our approach on real data with an experiment showing how the functional signal in subcortical brain regions of interest (ROIs) can be significantly predicted with genome-wide genotypes. In section 2, we introduce methodological prerequisites, then we describe our generic distributed machine learning approach for neuroimaging-genetic investigations and we present the cloud infrastructure. In section 3, we provide the description of the experiment and the results of the statistical analysis.

2. MATERIALS AND METHODS

2.1. NEUROIMAGING-GENETIC STUDY

Neuroimaging-genetic studies test the effect of genetic variables on imaging target variables in presence of exogenous variables. The imaging target variables are activation images obtained through functional Magnetic Resonance Imaging (fMRI), that yield a standardized effect related to experimental stimulation at each brain location of a reference brain space. For a study involving n subjects, we generally consider the following model:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta_1} + \boldsymbol{Z}\boldsymbol{\beta_2} + \boldsymbol{\epsilon},$$

where $\boldsymbol{Y}$ is a $n \times p$ matrix representing the signal of n subjects described each by p descriptors (e.g., voxels or ROIs of an fMRI contrast image), $\boldsymbol{X}$ is the $n \times q_1$ set of q_1 explanatory variables and $\boldsymbol{Z}$ the $n \times q_2$ set of q_2 covariates that explain some portion of the signal but are not to be tested for an effect. $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ are the fixed coefficients of the model to be estimated, and $\boldsymbol{\epsilon}$ is some Gaussian noise. $\boldsymbol{X}$ contains genetic measurements and variables in $\boldsymbol{Z}$ can be of any type (genetic, artificial, behavioral, experimental, ...).

2.1.1. The standard approach

It consists in fitting p Ordinary Least Square (OLS) regressions, one for each column of Y, as a target variable, and each time perform a statistical test (e.g., F-test) and interpret the results in term of significance (p-value). This approach suffers from some limitations. First, due to a low signal-to-noise ratio and a huge number of tests, this approach is not sensitive. Moreover, the statistical score only reflects the univariate correlation between a target and a set of q_1 explanatory variables, it does not inform on their predictive power when considered jointly. Secondly, with neuroimaging data as a signal, we are not in a *case vs. control* study. It raises the question whether the variability in a population can be imputed to few rare genetic variants or if it is the addition of many small effects of common variants. Unfortunately, the model holds only if $n \gg (q_1 + q_2)$, which is not the case with genome-wide genotypes.

2.1.2. Heritability assessment

The goal of our analysis is to estimate the proportion of differences in a trait between individuals due to genetic variability. Heritability evaluation traditionally consists in studying and comparing homozygous and dizygous twins, but recently it has been shown that it can be estimated using genome-wide genotypes (Lee et al., 2011; Lippert et al., 2011; Yang et al., 2011b). For instance, common variants are responsible of a large portion of the heritability of human height (Yang et al., 2010) or schizophrenia (Lee et al., 2012). These results show that the variance explained by each chromosome is proportional to its length. As we consider fMRI measurements in an unsupervised setting (no disease), this suggests to use regression models that do not enforce sparsity. Like the standard approach, heritability has some limitations. In particular, the estimation of heritability requires large sample sizes to have an acceptable standard error (at least 4000 according to Lee et al., 2012). Secondly, the heritability is the ratio between the variance of the trait and the genetic variance in a population. Therefore, for a given individual, a trait with an heritability at 0.6 does not mean it can be predicted at 60% on average with the genotype. It means that a fraction of the phenotype variability is simply explained by the average genetic structure of the population of interest.

2.1.3. *High-dimensional statistics*

The key point of our approach is to fit a model on training data (train set) and evaluate its goodness on unseen data (test set). To stabilize the impact of the sets for training and testing, a cross-validation loop is performed, yielding an average prediction score over the folds. This score yields a statistic value and a permutation test is performed to tabulate the distribution of this statistic under the null hypothesis and to estimate its significance (p-value). In practice, this corresponds to swapping the labels of the observations. As a prediction metric we generally choose the coefficient of determination (R^2), which is the ratio between the variance of the prediction and the variance of the phenotypes in the test set. If we consider all the genotypes at the same time, this approach is clearly related to *heritability*, but focuses on the predictive power of the model and its significance. Through cross-validation, the estimation of the $CV\text{-}R^2$ with an acceptable standard error does not require as large sample sizes as for the estimation of heritability (Yang et al., 2011a).

$$CV\text{-}R^2 = 1 - mean_{(train,\, test) \in \mathrm{split(n)}} \frac{\|\boldsymbol{Y}^{test} - \boldsymbol{X}^{test}\boldsymbol{\beta}_1^{train} - \boldsymbol{Z}^{test}\boldsymbol{\beta}_2^{train}\|^2}{\|\boldsymbol{Y}^{test} - \boldsymbol{Z}^{test}\boldsymbol{\beta}_2^{train}\|^2}$$

2.2. GENERIC PROCEDURE FOR DISTRIBUTED MACHINE LEARNING

If one just wants to compute the prediction score for few phenotypes, a multicore machine should be enough. But, if one is interested in the significance of this prediction score, one will probably need a computers farm (cloud, HPC cluster, etc.) Our approach consists in unifying the description and the computation for neuroimaging-genetic studies to scale from the desktop computer to the supercomputing facilities. The description of the statistical inference is provided by a descriptive configuration in human-readable and standard format: JSON (JavaScript Object Notation). This format requires no programming skills and is far easier to process as compared to the XML (eXtensible Markup Language) format. In a sense, our approach extends the Scikit-learn library (cf. next paragraph) for distributed computing, but focuses on a certain kind of inferences for neuroimaging-genetic studies. The next paragraphs describe the strategy, framework and implementation used to meet the heritability assessment objective.

2.2.1. *Scikit-learn*

Scikit-learn is a popular machine learning library in Python (Pedregosa et al., 2011) designed for a multicore station. In the Scikit-learn vocabulary, an `estimator` is an object that implements a `fit` and a `predict` method. For instance a `Ridge` object (lines 12–13 of **Figure 1**) is an `estimator` that computes the coefficients of the ridge regression model on the train set and uses these coefficients to predict data from the test set. If this object has a `transform` method, it is called a `transformer`. For instance a `SelectKbest` object (lines 10–11 of **Figure 1**) is a `transformer` that modifies the input data (the design matrix $\boldsymbol{X}$) by returning the K best explanatory variables w.r.t. a scoring function. Scikit-learn defines a `Pipeline` (lines 8–13 of **Figure 1**) as the combination of several `transformers` and an final `estimator`: It creates a combined estimator. Model selection procedures are provided to evaluate with a cross-validation the performance of an estimator (e.g., `cross_val_score`) or to select parameters on a grid (e.g., `GridSearchCV`).

2.2.2. *Permutations and covariates*

Standard machine learning procedures have not been designed to deal with covariates (such as those assembled in the matrix $\boldsymbol{Z}$), which have to be considered carefully in a permutation test (Anderson and Robinson, 2001). For the original data, we fit an Ordinary Least Square (OLS) model between $\boldsymbol{Y}$ and $\boldsymbol{Z}$, then we consider the residuals of the regression (denoted $R_{Y|Z}$) as the target for the machine learning estimator. For the permutation test, we permute $R_{Y|Z}$ (the permuted version is denoted $R_{Y|Z^*}$), then we fit an OLS model between $R_{Y|Z^*}$ and $\boldsymbol{Z}$, and we consider the residuals as the target for the estimator (Anderson and Robinson, 2001). The goal of the second OLS on the permuted residuals is to provide an optimal approximation (in terms of bias and computation) of the exact permutation tests while working on the reduced model.

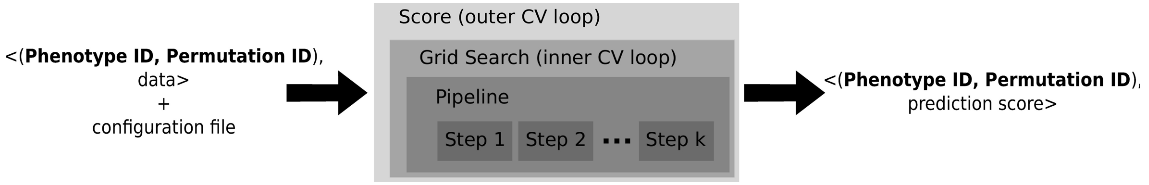
2.2.3. *Generic problem*

We identify a scheme common to the different kinds of inference that we would like to perform. For each target phenotype we want to compute a prediction score in the presence of covariates or not and to evaluate its significance with a permutation test. Scikit-learn algorithms are able to execute on multiple CPU cores, notably cross-validation loop, so a task will be executed on a multicore machine: cluster nodes or multicore virtual machine (VM). As the computational burden of different machine learning algorithms is highly variable, owing to the number of samples and the dimensionality of the data, we thus have to tune the number of tasks and their average computation time. An optimal way to tune the amount of work is to perform several permutations on the same data in a given task to avoid I/O bottlenecks. Finally, we put some constraints on the description of the machine learning estimator and the cross validation scheme:

- The prediction score is computed using the Scikit-learn `cross_val_score` function and the folds for this cross validation loop are generated with a `ShuffleSplit` object.
- An estimator is described with a Scikit-learn `pipeline` with one or more steps.
- Python can dynamically load modules such that a program can execute functions that are passed in a string or a configuration file. To notify that a string contains a Python module and an object or function to load, we introduce the prefix `DYNAMIC_IMPORT::`
- To select the best set of parameters for an estimator, model selection is performed using Scikit-learn `GridSearchCV` and a 5-folds inner cross-validation loop.

2.2.4. *Full example (cf. script in Figure 1)*

- *General parameters (Lines 1–3)*: The model contains covariates, the permutation test makes 10,000 iterations and only one permutation is performed in a task. 10,000 tasks per brain target phenotypes will be generated.
- *Prediction score (Lines 4–7)*: The metrics for the cross-validated prediction score is R^2, the cross-validation loop makes 10


<(Phenotype ID, Permutation ID),
data>
+
configuration file
Score (outer CV loop)
Grid Search (inner CV loop)
Pipeline
Step 1
Step 2
...
Step k
<(Phenotype ID, Permutation ID),
prediction score>

```
{"extract_cov": true,                                                    1
 "n_perm_total": 10000,                                                  2
 "n_perm_per_mapper": 1,                                                 3
 "cross_val_score": {                                                    4
    "score_func": "DYNAMIC_IMPORT::sklearn.metrics.r2_score"},           5
 "ShuffleSplit": {                                                       6
    "test_size": 0.2, "random_state": 0, "n_iter": 10},                  7
 "pipeline": [                                                           8
   ["FastFilterColinear", "gstat.data.utils.FastFilterColinear", {}],    9
   ["SelectKBest", "sklearn.feature_selection.SelectKBest", {           10
      "score_func": "DYNAMIC_IMPORT::sklearn.feature_selection.f_regression"}],11
   ["Ridge", "sklearn.linear_model.Ridge", {                            12
      "fit_intercept": true}]],                                         13
 "GridSearchCV": ["sklearn.grid_search.GridSearchCV", {}, [{            14
    "SelectKBest__k" : [10, 100, 1000],                                 15
    "Ridge__alpha" : [0.0001, 0.001, 0.01, 0.1, 1.]}]]}                 16
```

FIGURE 1 | Top: Representation of the computational framework: given the data, a permutation and a phenotype index together with a configuration file, a set of computations are performed, that involve two layers of cross-validation for setting the hyper-parameters and evaluate the accuracy of the model. This yields a statistical score associated with the given phenotype and permutation. **Bottom**: Example of complex configuration file that describes this set of operations. *General parameters (Lines 1–3)*: The model contains covariates, the permutation test makes 10,000 iterations and only one permutation is performed in a task. *Prediction score (Lines 4–7)*: The metrics for the cross-validated prediction score is R^2, the cross-validation loop makes 10 iterations, 20% of the data are left out for the test set and the seed of the random generator was set to 0. *Estimator pipeline (Lines 8–13)*: The first step consists in filtering collinear vectors, the second step selects the *K* best features and the final step is a ridge estimator. *Parameters selection (Lines 14–16)*: Two parameters of the estimator have to be set: the *K* for the *SelectKBest* and the *alpha* of the *Ridge* regression. A set of 3 × 5 parameters are evaluated.

iterations, 20% of the data are left out for the test set and the seed of the random generator was set to 0.

- *Estimator pipeline (Lines 8–13)*: The first step consist in filtering collinear vectors, the second step selects the *K* best features and the final step is a ridge estimator.
- *Parameters selection (Lines 14–16)*: Two parameters of the estimator have to be set: the *K* for the `SelectKBest` and the *alpha* of the `Ridge` regression. A set of 3 × 5 parameters are evaluated.

2.3. THE CLOUD COMPUTING ENVIRONMENT

Although researchers have relied mostly on their own clusters or grids, clouds are raising an increasing interest (Jackson et al., 2010; Simmhan et al., 2010; Ghoshal et al., 2011; Hiden et al., 2012; Juve et al., 2012). While shared clusters or grids often imply a quota-based usage of the resources, those from clouds are owned until they are explicitly released by the user. Clouds are easier to use since most of the details are hidden to the end user (e.g., network physical implementation). Depending on the characteristics of the targeted problem, this is not always an advantage (e.g., collective communications). Last but not least, clouds avoid owning expensive infrastructures—and associated high cost for buying and operating—that require technical expertise.

The cloud infrastructure is composed of multiple data centers, which integrate heterogeneous resources that are exploited seamlessly. For instance, the Windows Azure cloud has five sites in United States, two in Europe and three in Asia. As resources are granted *on-demand*, the cloud gives the illusion of infinite resources. Nevertheless, cloud data centers face the same load problems (e.g., workload balancing, resource idleness, etc.) as traditional grids or clusters.

In addition to the computation capacity, clouds often provide data-related services, like object storage for large datasets (e.g., S3 from Amazon or Windows Azure Blob) and queues for short message communication.

2.4. NEUROIMAGING-GENETICS COMPUTATION IN THE CLOUD

In practice, the workload of the A-Brain application [1] is more resource demanding than the typical cloud applications and could induce two undesirable situations: (1) other clients do not have enough resource to lease on-demand in a particular data center; (2) the computation creates performance degradations for other applications in the data center (e.g., by occupying the network bandwidth, or by creating high number of concurrent requests on the cloud storage service). Therefore, we divide the workload into smaller sub-problems and we select the different datacenters in collaboration with the cloud provider.

[1] http://www.irisa.fr/kerdata/abrain/

For balancing the load of the A-Brain application, the computation was distributed across four *deployments* in the two biggest Windows Azure datacenters. In the cloud context, a *deployment* denotes a set of leased resources, which are presented to the user as a set of uniform machines, called *compute nodes*. Each deployment is independent and isolated from the other deployments. When a compute node starts, the user application is automatically uploaded and executed. The compute nodes of a deployment belong to the same virtual private network and communicate with the outside world or other deployments either through *public endpoints* or using the cloud storage services (i.e., Windows Azure Blob or Queue).

TomusBlobs (Costan et al., 2013) is a data management system designed for concurrency-optimized PaaS-level (Platform as a Service) cloud data management. The system relies on the available local storage of the compute nodes in order to share input files and save output files. We built a processing framework (called TomusMapReduce) derived from MapReduce (Chu et al., 2006; Dean and Ghemawat, 2008) on top of TomusBlobs, such that it leverages its benefits by collocating data with computation. Additionally, the framework is restricted to *associative* and *commutative* reduction procedures (Map-IterativeReduce model Tudoran et al., 2012) in order to allow efficient out-of-order and parallel processing for the reduce phase. Although MapReduce is designed for single cluster processing, the latter constraint enables straightforward geographically distributed processing. The hierarchical MapReduce (which is described in Costan et al., 2013) aggregates several deployments with *MapReduce engines* and a last deployment that contains a *MetaReducer*, that computes the final result, and a *Splitter*, that partitions the data and manages the overall workload in order to leverage data locality. Job descriptions are sent to the MapReduce engines via Windows Azure Queue and the MetaReducer collects intermediate results via Windows Azure Blob. For our application, we use the Windows Azure Blob storage service instead of TomusBlobs for several reasons: (1) concurrency-optimized capabilities are not relevant here; (2) for a very long run, it is better to rely on a proven storage; (3) TomusBlob storage does not support yet multi-deployments setting. An overview of the framework is shown in **Figure 2**.

For our application, the *Map* step yields a prediction score for an image phenotype and a permutation, while the *reduce* step consists in collecting all results to compute statistic distribution and corrected *p*-values. The reduce operation is trivially commutative and associative as it consists in searching the maximum of the statistic for each permutation (Westfall and Young, 1993). The upper part of **Figure 1** gives an overview of the generic mapper.

2.5. IMAGEN: A NEUROIMAGING-GENETIC DATASET

IMAGEN is a European multi-centric study involving adolescents (Schumann et al., 2010). It contains a large functional neuroimaging database with fMRI associated with 99 different contrast images for 4 protocols in more than 2000 subjects, who gave informed signed consent. Regarding the functional neuroimaging data, we use the Stop Signal Task protocol (Logan, 1994) (SST), with the activation during a *[go wrong]* event, i.e., when the subject pushes the wrong button. Such an experimental contrast is likely to show complex mental processes (inhibition failure, *post-hoc* emotional reaction of the subject), that may be hard to disentangle. Our expectation is that the amount of Blood Oxygen-Level Dependent (BOLD) response associated with such events provides a set of global markers that may reveal some heritable psychological traits of the participants. Eight different 3T scanners from multiple manufacturers (GE, Siemens, Philips) were used to acquire the data. Standard preprocessing, including slice timing correction, spike and motion correction, temporal detrending (functional data) and spatial normalization (anatomical and functional data), were performed using the SPM8 software and its default parameters; functional images were resampled at 3 mm resolution. All images were warped in the MNI152 coordinate space. Obvious outliers detected using simple rules such as large registration or segmentation errors or very large motion parameters were removed after this step. BOLD time series was recorded using Echo-Planar Imaging, with $TR = 2200$ ms, $TE = 30$ ms, flip angle $= 75°$ and spatial resolution $3 \times 3 \times 3$ mm. Gaussian smoothing at 5 mm-FWHM

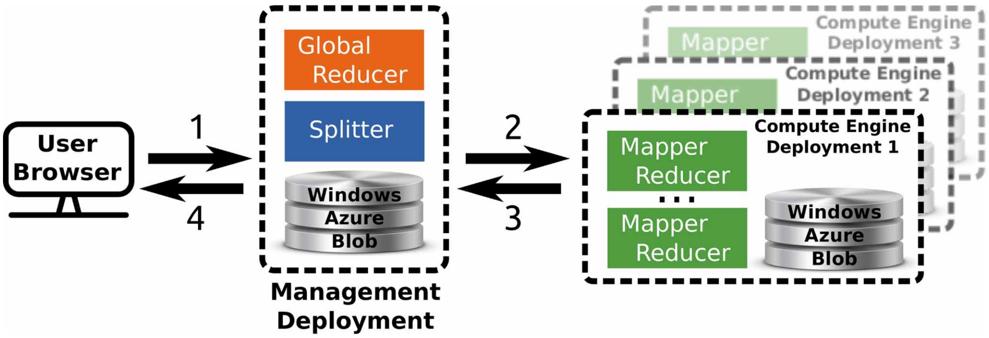


FIGURE 2 | Overview of the multi site deployment of a hierarchical Tomus-MapReduce compute engine. **(1)** The end-user uploads the data and configures the statistical inference procedure on a webpage. **(2)** The *Splitter* partitions the data and manages the workload. The compute engines retrieves job information trough the Windows Azure Queues. **(3)** Compute engines perform the *map* and *reduce* jobs. The management deployment is informed of the progression via the Windows Azure Queues system and thus can manage the execution of the *global reducer*. **(4)** The user downloads the results of the computation on the webpage of the experiment.

was finally added. Contrasts were obtained using a standard linear model, based on the convolution of the time course of the experimental conditions with the canonical hemodynamic response function, together with standard high-pass filtering procedure and temporally auto-regressive noise model. The estimation of the first-level was carried out using the SPM8 software. T1-weighted MPRAGE anatomical images were acquired with spatial resolution $1 \times 1 \times 1$ mm, and gray matter probability maps were available for 1986 subjects as outputs of the SPM8 *New Segmentation* algorithm applied to the anatomical images. A mask of the gray matter was built by averaging and thresholding the individual gray matter probability maps. More details about data preprocessing can be found in Thyreau et al. (2012).

DNA was extracted from blood samples using semi-automated process. Genotyping was performed genome-wide using Illumina Quad 610 and 660 chips, yielding approximately 600,000 autosomic SNPs. 477,215 SNPs are common to the two chips and pass *plink* standard parameters (Minor Allele Frequency >0.05, Hardy-Weinberg Equilibrium $P < 0.001$, missing rate per SNP <0.05).

3. AN APPLICATION AND RESULTS

3.1. THE EXPERIMENT

The aim of this experiment is to show that our framework has the potential to explore links between neuroimaging and genetics. We consider an fMRI contrast corresponding to events where subjects make motor response errors (*[go wrong]* fMRI contrast from a Stop Task Signal protocol). Subjects with too many missing voxels or with bad task performance were discarded. Regarding genetic variants, 477,215 SNPs were available. Age, sex, handedness and acquisition center were included in the model as confounding variables. Remaining missing data were replaced by the median over the subjects for the corresponding variables. After applying all exclusion criteria 1459 subjects remained for analysis. Analyzing the whole brain with all the genetic variants remains intractable due to the time and memory requirements and dimension reduction techniques have to be employed.

3.1.1. Prior neuroimaging dimension reduction

In functional neuroimaging, brain atlases are mainly used to provide a low-dimensional representation of the data by considering signal averages within groups of neighboring voxels. In this experiment we focus on the subcortical nuclei using the Harvard–Oxford subcortical atlas. We extract the functional signal of 14 regions of interest, 7 in each hemisphere: thalamus, caudate, putamen, pallidum, hippocampus, amygdala and accumbens (see **Figure 4**). White matter, brain stem and ventricles are of no interest for functional activation signal and were discarded. This prior dimension reduction decreases the number of phenotypes from more than 50,000 voxels to 14 ROIs.

3.1.2. Configuration used (cf. script in Figure 3)

- *(Lines 1–3)*: covariates, 10,000 permutations and 5 permutations per computation unit (mapper).
- *(Lines 4–7)*: 10-folds cross-validated R^2.
- *(Lines 9–11)*: The first step of the pipeline is an univariate features selection ($K = 50,000$). This step is used as a dimension reduction so that the next step fits in memory.
- *(Lines 12–13)*: The second and last step is the ridge estimator with a low penalty ($alpha = 0.0001$).

The goal of the experiment described by this configuration file is to evaluate how the 50,000 mostly correlated genetic variants, once taken together, are predictive of each ROI and to associate a *p*-value with these prediction scores. Note that more than 50,000 covariates would not fit into memory. This configuration generates 28,000 map tasks ($14 \times 10,000/5$), but we can set to 1 the number of permutations per task, which means that the computation can use up to 140,000 multicore computers in parallel, and thus millions of CPU cores.

3.1.3. The cloud experimental setup

The experiment was performed using the Microsoft Windows Azure PaaS cloud in the North and West US datacenters, that were recommended by the Microsoft team for their capacity. We use the Windows Azure storage services (Blob and Queue) in both datacenters in order to take advantage of the data locality. Due to our memory requirements, the *Large VM* type (4 CPU cores, 7 GB of memory and 1000 GB of disk) is the best fit regarding the Azure VMs offer[2].

3.1.4. TomusBlobs

We set up two deployments in each of the two recommended sites for a total of four deployments. It used 250 large VM nodes, totalizing 1000 CPUs: each of the 3 MapReduce engines deployments had 82 nodes and the last deployment used 4 nodes. The reduction process was distributed in approximately 600 reduce jobs.

3.2. RESULTS

3.2.1. Cloud aspects

The experiment timespan was 14 days. The processing time for a single map job is approximately 2 h. There are no noticeable time differences between the execution times of the map jobs with respect to the geographical location. In large infrastructures like the clouds, failures are possible and applications need to cope with this. In fact, during the experiment the Azure services became temporary inaccessible [3], due to a failure of a secured certificate. Despite this problem, the framework was able to handle the failure with a fault tolerance mechanism which suspended the computation until all Azure services became available again. The monitoring mechanism of the *Splitter*, that supervises the computation progress, was able to restore aborted jobs. The IterativeReduce approach eliminates the implicit barrier between mappers and reducers, but yields negligible gains due to the huge workload of the mappers. The effective cost of the experiment was approximately equal to 210,000 h of sequential computation, which corresponds to almost $20,000 (VM pricing, storage and outbound traffic).

[2] http://msdn.microsoft.com/fr-fr/library/windowsazure/dn197896.aspx

[3] Azure Failure Incident: http://readwr.it/tAq

```
{"extract_cov": true,                                                    1
 "n_perm_total": 10000,                                                  2
 "n_perm_per_mapper": 5,                                                 3
 "cross_val_score": {                                                    4
    "score_func": "DYNAMIC_IMPORT::sklearn.metrics.r2_score"},           5
 "ShuffleSplit": {                                                       6
    "test_size": 0.2, "random_state": 0, "n_iter": 10},                  7
 "pipeline": [                                                           8
   ["SelectKBest", "sklearn.feature_selection.SelectKBest", {            9
      "score_func": "DYNAMIC_IMPORT::gstat.stats.utils.f_regression",   10
      "k": 50000}],                                                     11
   ["Ridge", "sklearn.linear_model.Ridge", {                            12
      "fit_intercept": true, "alpha": 0.0001}]]}                        13
```

FIGURE 3 | Configuration used for the experiment. *(Lines 1–3)*: Covariates, 10,000 permutations and five permutations per computation unit (mapper). *(Lines 4–7)*: 10-folds cross-validated R^2. *(Lines 9–11)*: The first step of the pipeline is an univariate features selection ($K = 50, 000$). This step is used as a dimension reduction so that the next step fits in memory. *(Lines 12–13)*: The second and last step is the ridge estimator with a low penalty (*alpha* = 0.0001).

ROI name		CV-R^2	fwe corr. p-value
Thalamus	left	**0.026**	1.10^{-4}
	right	**0.038**	1.10^{-4}
Caudate	left	**0.003**	2.10^{-4}
	right	-0.012	3.10^{-4}
Putamen	left	**0.019**	1.10^{-4}
	right	**0.006**	2.10^{-4}
Pallidum	left	**0.018**	1.10^{-4}
	right	-0.010	3.10^{-4}
Hippocampus	left	**0.010**	2.10^{-4}
	right	**0.020**	1.10^{-4}
Amygdala	left	**0.016**	1.10^{-4}
	right	**0.015**	1.10^{-4}
Accumbens	left	**0.022**	1.10^{-4}
	right	-0.002	2.10^{-4}

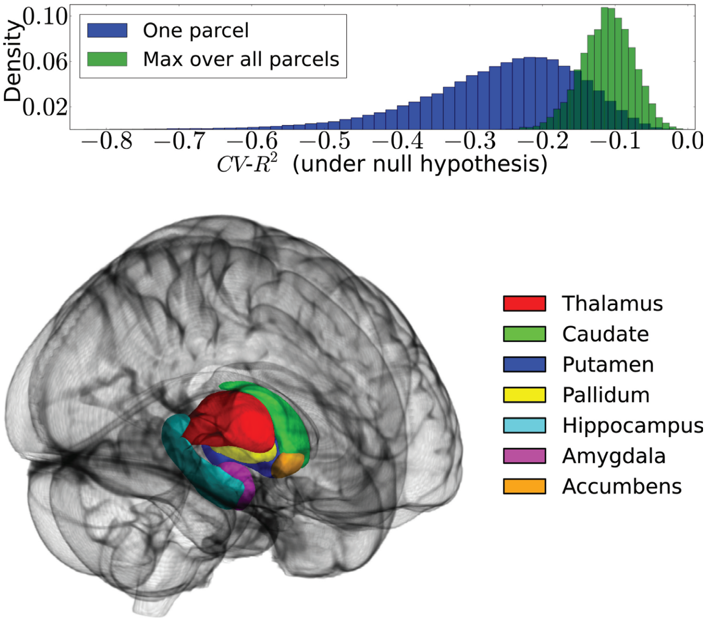


FIGURE 4 | Results of the real data analysis procedure. (**Left**) predictive accuracy of the model measured by cross-validation, in the 14 regions of interest, and associated statistical significance obtained in the permutation test. (**Up right**) distribution of the *CV-R*^{2} at chance level, obtained through a permutation procedure. The distribution of the max over all ROIs is used to obtain the family-wise error corrected significance of the test. (**Bottom right**) outline of the chosen ROIs.

3.2.2. Application side

Figure 4 shows a summary of the results. Despite the fact that some prediction scores are negative, the activation signal in each ROI is fit significantly better than chance using the 50,000 best genetic variants over the 477,215. The mean BOLD signal is better predicted in the left and right thalamus. The distribution of the CV-R^2 is also very informative, showing that by chance the mean prediction score is negative (familywise-error corrected or not). While this phenomenon is somewhat counter-intuitive within the framework of classical statistics, it should be pointed out that the cross-validation procedure used here opens the possibility of negative R^2: this quantity is by definition a model comparison statistic that takes the difference between a regression model with a non-informative model; in high-dimensional settings, a poorly fitting linear model performs (much) worse than a non-informative model. Hence a model performing at chance gets a negative score: This is actually what happens systematically when the association between y and X is broken by the permutation procedure, even if we consider the supremum over many statistical tests (Westfall and Young, 1993). A slightly negative value can thus be the marker of a significant association between the variables of interest. Twin and SNP-based studies suggest high heritability of structural brain measures, such as total amount of gray and white matter, overall brain volume and addiction-relevant subcortical regions. Heritability estimates for brain measures are as high as 0.89 (Kremen et al., 2010) or even up to 0.96 (van Soelen et al., 2012) and subcortical regions appear to be moderately to highly heritable. One recent study on subcortical volumes (den Braber et al., 2013) reports highest heritability estimates for the thalamus (0.80) and caudate nucleus (0.88) and lowest for the left nucleus accumbens (0.44). Despite the fact that the $CV\text{-}R^2$ metric is not exactly an heritability measurement, our metric evaluates the predictability of the fitted model (i.e., how well it predicts the activation signal of a brain region with genetic measurements on unseen data) which is a good proxy for heritability. Thus, our results confirm that brain activation signals are an heritable feature in subcortical regions. These experiments can be used as a basis to further localize the genetic regions (pathways or genes) that are actually predictive of the functional activation. An important extension of the present work is clearly to extend this analysis to the cortical regions.

4. CONCLUSION

The quantitative evaluation of statistical models with machine learning techniques represents an important step in the comprehension of the associations between brain image phenotypes and genetic data. Such approaches require cross validation loops to set the hyper-parameters and to evaluate performances. Permutations have to be used to assess the statistical significance of the results, thus yielding prohibitively expensive analyses. In this paper, we present a framework that can deal with such a computational burden. It relies on two key points: (1) it wraps the Scikit-learn library to enable coarse grain distributed computation. Yet it enforces some restrictions, i.e., it solves only a given class of problems (pipeline structure, cross-validation procedure and permutation test). The result is a simple generic code (few lines) that provides the user a quick way to conduct early, small-scale investigations on its own computer or at a larger scale on a high-performance computing cluster. With JSON we provide a standard format for the description of statistical inference so that no programming skills are required and so that it can be easily generated from a webpage form. (2) TomusBLOB permits to execute seamlessly the very same code on the Windows Azure cloud. We could also disable some parts of TomusBLOB to achieve a good compromise between the capabilities and the robustness. We demonstrate the scalability and the efficiency of our framework with a 2 weeks geographically distributed execution on hundreds of virtual machines. The results confirm that brain activation signals are an heritable feature.

ACKNOWLEDGMENTS

This work was supported primarily by the Microsoft INRIA joint centre grant *A-brain* and secondarily by the Digiteo *ICoGeN* grant and the ANR grant ANR-10-BLAN-0128. HPC investigations were carried out using the computing facilities of the CEA-DSV and CATI cluster. The data were acquired within the IMAGEN project, which receives research funding from the E.U. Community's FP6, LSHM-CT-2007-037286. This manuscript reflects only the author's views and the Community is not liable for any use that may be made of the information contained therein.

REFERENCES

Anderson, M. J., and Robinson, J. (2001). Permutation tests for linear models. *Aust. N. Z. J. Stat.* 43, 75–88. doi: 10.1111/1467-842X.00156

Bunea, F., She, Y., Ombao, H., Gongvatana, A., Devlin, K., and Cohen, R. (2011). Penalized least squares regression methods and applications to neuroimaging. *Neuroimage* 55, 1519–1527. doi: 10.1016/j.neuroimage.2010.12.028

Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G. R., Ng, A. Y., et al. (2006). "Map-reduce for machine learning on multicore," in *NIPS* (Vancouver, BC), 281–288.

Costan, A., Tudoran, R., Antoniu, G., and Brasche, G. (2013). TomusBlobs: scalable data-intensive processing on Azure clouds. *J. Concurr. Comput. Pract. Exp.* doi: 10.1002/cpe.3034

Da Mota, B., Frouin, V., Duchesnay, E., Laguitton, S., Varoquaux, G., Poline, J.-B., et al. (2012). "A fast computational framework for genome-wide association studies with neuroimaging data," in *20th International Conference on Computational Statistics* (Limassol).

Dean, J., and Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 107–113. doi: 10.1145/1327452.1327492

den Braber, A., Bohlken, M. M., Brouwer, R. M., van 't Ent, D., Kanai, R., Kahn, R. S., et al. (2013). Heritability of subcortical brain measures: a perspective for future genome-wide association studies. *Neuroimage* 83C, 98–102. doi: 10.1016/j.neuroimage.2013.06.027

Floch, E. L., Guillemot, V., Frouin, V., Pinel, P., Lalanne, C., Trinchera, L., et al. (2012). Significant correlation between a set of genetic polymorphisms and a functional brain network revealed by feature selection and sparse partial least squares. *Neuroimage* 63, 11–24. doi: 10.1016/j.neuroimage.2012.06.061

Gao, X., Becker, L. C., Becker, D. M., Starmer, J. D., and Province, M. A. (2010). Avoiding the high Bonferroni penalty in genome-wide association studies. *Genet. Epidemiol.* 34, 100–105. doi: 10.1002/gepi.20430

Ge, T., Feng, J., Hibar, D. P., Thompson, P. M., and Nichols, T. E. (2012). Increasing power for voxel-wise genome-wide association studies: the random field theory, least square kernel machines and fast permutation procedures. *Neuroimage* 63, 858–873. doi: 10.1016/j.neuroimage.2012.07.012

Ghoshal, D., Canon, R. S., and Ramakrishnan, L. (2011). "I/o performance of virtualized cloud environments," in *Proceedings of the Second International Workshop on Data Intensive Computing in the Clouds*, DataCloud-SC '11 (New York, NY: ACM), 71–80. doi: 10.1145/2087522.2087535

Hibar, D. P., Stein, J. L., Kohannim, O., Jahanshad, N., Saykin, A. J., Shen, L., et al. (2011). Voxelwise gene-wide association study (vGeneWAS): multivariate gene-based association testing in 731 elderly subjects. *Neuroimage* 56, 1875–1891. doi: 10.1016/j.neuroimage.2011.03.077

Hiden, H., Woodman, S., Watson, P., and Cala, J. (2012). Developing cloud applications using the e-science central platform. *Proc. R. Soc. A.* 371:20120085. doi: 10.1098/rsta.2012.0085

Jackson, K. R., Ramakrishnan, L., Runge, K. J., and Thomas, R. C. (2010). "Seeking supernovae in the clouds: a performance study," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10 (New York, NY: ACM), 421–429. doi: 10.1145/1851476.1851538

Juve, G., Deelman, E., Berriman, G. B., Berman, B. P., and Maechling, P. (2012). An evaluation of the cost and performance of scientific workflows on amazon ec2. *J. Grid Comput.* 10, 5–21. doi: 10.1007/s10723-012-9207-6

Kohannim, O., Hibar, D. P., Stein, J. L., Jahanshad, N., Jackand, C. R. Jr., Weiner, M. W., et al. (2011). "Boosting power to detect genetic associations in imaging using multi-locus, genome-wide scans and ridge regression," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (Chicago, IL), 1855–1859. doi: 10.1109/ISBI.2011.5872769

Kremen, W. S., Prom-Wormley, E., Panizzon, M. S., Eyler, L. T., Fischl, B., Neale, M. C., et al. (2010). Genetic and environmental influences on the size of specific brain regions in midlife: the vetsa mri study. *Neuroimage* 49, 1213–1223. doi: 10.1016/j.neuroimage.2009.09.043

Lee, S. H., DeCandia, T. R., Ripke, S., Yang, J., Schizophrenia Psychiatric Genome-Wide Association Study Consortium (PGC-SCZ), and International Schizophrenia Consortium, et al. (2012). Estimating the proportion of variation in susceptibility to schizophrenia captured by common snps. *Nat. Genet.* 44, 247–250. doi: 10.1038/ng.1108

Lee, S. H., Wray, N. R., Goddard, M. E., and Visscher, P. M. (2011). Estimating missing heritability for disease from genome-wide association studies. *Am. J. Hum. Genet.* 88, 294–305. doi: 10.1016/j.ajhg.2011.02.002

Lippert, C., Listgarten, J., Liu, Y., Kadie, C. M., Davidson, R. I., and Heckerman, D. (2011). Fast linear mixed models for genome-wide association studies. *Nat. Methods* 8, 833–835. doi: 10.1038/nmeth.1681

Logan, G. D. (1994). On the ability to inhibit thought and action: a users' guide to the stop signal paradigm. *Psychol. Rev.* 91, 295–327. doi: 10.1037/0033-295X.91.3.295

Meinshausen, N., and Bühlmann, P. (2010). Stability selection. *J. R. Stat. Soc. B (Stat. Methodol.)* 72, 417–473. doi: 10.1111/j.1467-9868.2010.00740.x

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.

Schumann, G., Loth, E., Banaschewski, T., Barbot, A., Barker, G., Büchel, C., et al. (2010). The imagen study: reinforcement-related behaviour in normal brain function and psychopathology. *Mol. Psychiatry* 15, 1128–1139. doi: 10.1038/mp.2010.4

Simmhan, Y., van Ingen, C., Subramanian, G., and Li, J. (2010). "Bridging the gap between desktop and the cloud for escience applications," in *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, CLOUD '10 (Washington, DC: IEEE Computer Society), 474–481. doi: 10.1109/CLOUD.2010.72

Stein, J. L., Hua, X., Lee, S., Ho, A. J., Leow, A. D., Toga, A. W., et al. (2010). Voxelwise genome-wide association study (vGWAS). *Neuroimage* 53, 1160–1174. doi: 10.1016/j.neuroimage.2010.02.032

Thyreau, B., Schwartz, Y., Thirion, B., Frouin, V., Loth, E., Vollstädt-Klein, S., et al. (2012). Very large fMRI study using the IMAGEN database: sensitivity-specificity and population effect modeling in relation to the underlying anatomy. *Neuroimage* 61, 295–303. doi: 10.1016/j.neuroimage.2012.02.083

Tudoran, R., Costan, A., and Antoniu, G. (2012). "Mapiterativereduce: a framework for reduction-intensive data processing on azure clouds," in *Proceedings of 3rd International Workshop on MapReduce and Its Applications Date*, MapReduce '12 (New York, NY: ACM), 9–16. doi: 10.1145/2287016.2287019

van Soelen, I. L. C., Brouwer, R. M., Peper, J. S., van Leeuwen, M., Koenis, M. M. G., van Beijsterveldt, T. C. E. M., et al. (2012). Brain scale: brain structure and cognition: an adolescent longitudinal twin study into the genetic etiology of individual differences. *Twin. Res. Hum. Genet.* 15, 453–467. doi: 10.1017/thg.2012.4

Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. (2008). A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.* 39, 50–55. doi: 10.1145/1496091.1496100

Vounou, M., Nichols, T. E., Montana, G., and Initiative, A. D. N. (2010). Discovering genetic associations with high-dimensional neuroimaging phenotypes: a sparse reduced-rank regression approach. *Neuroimage* 53, 1147–1159. doi: 10.1016/j.neuroimage.2010.07.002

Westfall, P. H., and Young, S. S. (1993). *Resampling-Based Multiple Testing : Examples and Methods for P-Value Adjustment*. New York, NY: Wiley.

Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., et al. (2010). Common snps explain a large proportion of the heritability for human height. *Nat. Genet.* 42, 565–569. doi: 10.1038/ng.608

Yang, J., Lee, S. H., Goddard, M. E., and Visscher, P. M. (2011a). Gcta: a tool for genome-wide complex trait analysis. *Am. J. Hum. Genet.* 88, 76–82. doi: 10.1016/j.ajhg.2010.11.011

Yang, J., Manolio, T. A., Pasquale, L. R., Boerwinkle, E., Caporaso, N., Cunningham, J. M., et al. (2011b). Genome partitioning of genetic variation for complex traits using common snps. *Nat. Genet.* 43, 519–525. doi: 10.1038/ng.823

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 31 October 2013; accepted: 17 March 2014; published online: 08 April 2014.
Citation: Da Mota B, Tudoran R, Costan A, Varoquaux G, Brasche G, Conrod P, Lemaitre H, Paus T, Rietschel M, Frouin V, Poline J-B, Antoniu G, Thirion B and IMAGEN Consortium (2014) Machine learning patterns for neuroimaging-genetic studies in the cloud. Front. Neuroinform. ***8****:31. doi: 10.3389/fninf.2014.00031*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 23 April 2014
doi: 10.3389/fninf.2014.00041

High-throughput neuroimaging-genetics computational infrastructure

Ivo D. Dinov[1,2,3]*, *Petros Petrosyan*[1], *Zhizhong Liu*[1], *Paul Eggert*[1,4], *Sam Hobel*[1], *Paul Vespa*[5], *Seok Woo Moon*[6], *John D. Van Horn*[1], *Joseph Franco*[1] and *Arthur W. Toga*[1,2]

[1] *Laboratory of Neuro Imaging, Institute for Neuroimaging and Informatics, University of Southern California, Los Angeles, CA, USA*
[2] *Biomedical Informatics Research Network, Information Sciences Institute, University of Southern California, Los Angeles, CA, USA*
[3] *Statistics Online Computational Resource, University of Michigan, UMSN, Ann Arbor, MI, USA*
[4] *Department of Computer Science, University of California, Los Angeles, Los Angeles, CA, USA*
[5] *Brain Injury Research Center, Department of Neurosurgery, David Geffen School of Medicine, University of California, Los Angeles, Los Angeles, CA, USA*
[6] *Department of Neuropsychiatry, Konkuk University School of Medicine, Seoul, Korea*

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Moritz A. Kirschmann, Institute for BioMedical Research, Basel
Dav Clark, University of California, USA
Timothy M. O'Keefe, Harvard University, USA
Randy Buckner, Harvard University, USA (in collaboration with Timothy M. O'Keefe)

Correspondence:
Ivo D. Dinov, Laboratory of Neuro Imaging, Institute for Neuroimaging and Informatics, 2001 North Soto Street – Room 102, MC 9232, Los Angeles, CA 90033, USA
e-mail: dinov@loni.usc.edu;
URL: http://www.loni.usc.edu

Many contemporary neuroscientific investigations face significant challenges in terms of data management, computational processing, data mining, and results interpretation. These four pillars define the core infrastructure necessary to plan, organize, orchestrate, validate, and disseminate novel scientific methods, computational resources, and translational healthcare findings. Data management includes protocols for data acquisition, archival, query, transfer, retrieval, and aggregation. Computational processing involves the necessary software, hardware, and networking infrastructure required to handle large amounts of heterogeneous neuroimaging, genetics, clinical, and phenotypic data and meta-data. Data mining refers to the process of automatically extracting data features, characteristics and associations, which are not readily visible by human exploration of the raw dataset. Result interpretation includes scientific visualization, community validation of findings and reproducible findings. In this manuscript we describe the novel high-throughput neuroimaging-genetics computational infrastructure available at the Institute for Neuroimaging and Informatics (INI) and the Laboratory of Neuro Imaging (LONI) at University of Southern California (USC). INI and LONI include ultra-high-field and standard-field MRI brain scanners along with an imaging-genetics database for storing the complete provenance of the raw and derived data and meta-data. In addition, the institute provides a large number of software tools for image and shape analysis, mathematical modeling, genomic sequence processing, and scientific visualization. A unique feature of this architecture is the Pipeline environment, which integrates the data management, processing, transfer, and visualization. Through its client-server architecture, the Pipeline environment provides a graphical user interface for designing, executing, monitoring validating, and disseminating of complex protocols that utilize diverse suites of software tools and web-services. These pipeline workflows are represented as portable XML objects which transfer the execution instructions and user specifications from the client user machine to remote pipeline servers for distributed computing. Using Alzheimer's and Parkinson's data, we provide several examples of translational applications using this infrastructure[1].

Keywords: aging, pipeline, neuroimaging, genetics, computation solutions, Alzheimer's disease, big data, visualization

INTRODUCTION

The long-term objectives of computational neuroscience research are to develop models, validate algorithms and engineer powerful tools facilitating the understanding of imaging, molecular, cellar, genetic, and environmental associations with brain circuitry and observed phenotypes. Most of the time, functioning teams of interdisciplinary investigators are necessary to develop innovative approaches to substantively expand the ways by which brain structure and function can be imaged in humans. Prototype development, proof of concept pilot studies and high-risk, high-impact research requires substantial infrastructure to support the data management, processing and collaboration.

There are significant barriers that inhibit our ability to understand the fundamental relations between brain states and the wide

[1] Some of the data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (http://adni.loni.usc.edu/). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

spectrum of observable, direct and indirect, biological, genetic, imaging, clinical, and phenotypic markers. Some of these challenges pertain to lack of models and algorithms for representing heterogeneous data, e.g., classifying normal and pathological variation (biological noise vs. technological errors) (Liu et al., 2012; Sloutsky et al., 2013). Others are driven by limitations in the available hardware and infrastructure resources, e.g., data size and complexity, data management, and sharing logistics, Distributed processing and data mining (Dinov et al., 2013; Kandel et al., 2013; Van Horn and Toga, 2013).

There are a number of teams and ongoing efforts that develop computational infrastructures to address specific research needs. For instance, the efforts of the Enhancing Neuroimaging Genetics through Meta-Analysis (ENIGMA) consortium (http://enigma.loni.usc.edu) represents a collection of World-wide research groups which agreed on a social networking strategy for data aggregation and sharing (Novak et al., 2012). ENIGMA manages imaging and genomics data facilitating the process of understanding brain structure and function using structural, functional, diffusion imaging and genome-wide association study (GWAS) data. The network's goal is to enable meta-research and replicated findings via increasing sample-sizes (cf. statistical power to detect phonotypic, imaging or genetic effects) in a collaborative fashion where investigators and groups share algorithms, data, information, and tools.

The high-throughput analysis of large amounts of data has become the ubiquitous norm in many computational fields, including neuroimaging (Barker and Van Hemert, 2008; Barrett et al., 2009; Dinov et al., 2009). The driving forces in this natural evolution of computerization and protocol automation are parallelization, increased network bandwidth, and the wide distribution of efficient and potent computational and communication resources. In addition, there are now more and larger data archives, often accumulating many hundreds, if not thousands, of subjects with enormous amounts of data. These can only be processed using efficient and structured systems. Efficient and effective tool interoperability is critical in many scientific endeavors as it enables new types of analyses, facilitates new applications, and promotes interdisciplinary collaborations (Dinov et al., 2008). The Pipeline Environment (Rex et al., 2003; Dinov et al., 2009) is a visual programming language and execution environment that enables the construction of complete study designs and management of data provenance in the form of complex graphical workflows. It facilitates the construction, validation, execution, and dissemination of analysis protocols, computational tools, and data services. The Pipeline has been used to construct advanced neuroimaging protocols analyzing multi-subject data derived from the largest publically available archives, including the International Consortium for Brain Mapping (ICBM) (Mazziotta et al., 1995), Alzheimer's Disease Neuroimaging Initiative (ADNI) (Mueller et al., 2005), Australian twin data of brain activation and heritability (Blokland et al., 2008), British infant database (Gousias et al., 2008), and the MNI (Evans, 2006) pediatric database.

Other significant efforts to provide computational infrastructure for high throughput brain data analyses include Taverna (Oinn et al., 2005) http://www.taverna.org.uk, Kepler (Ludäscher et al., 2006) kepeler-project.org, Khoros (Kubica et al., 1998), www.khoral.com, Trident Workbench (Toga et al., 2012), http://tridentworkflow.codeplex.com, Karma2 (Simmhan et al., 2008), http://www.extreme.indiana.edu/dist/java-repository/workflow-tracking/, Galaxy (Goecks et al., 2010), http://galaxy.psu.edu, and many others.

Examples of significant scientific, computational, and analytic challenges include:

1. *Software Tool Interoperability:* Differences in software development strategies can force intrinsic incompatibilities in algorithm design, implementation strategy, data format, or tool invocation syntax. For example, there are data type, array management, and processing differences in different language platforms, which complicate the integration of inputs and outputs. There also can be variations in implicit and explicit parameter specifications and services vs. command-line invocation syntax. The Distributed Pipeline addresses this barrier by providing an extensible markup language protocol for dynamic interoperability of diverse genomics data, informatics software tools, and web-services.
2. *Hardware Platform Dependencies:* Processor endianness (e.g., byte-swaps), architectural differences (e.g., 32 vs. 64-bit), compiler variations, and security incompatibilities cause significant problems in the integration of data and computational resources residing on multiple platforms. These hardware idiosyncrasies limit the potential to utilize the most appropriate computational resources on multi-platform systems and reduce the efficiency of many computational approaches. The distributed Pipeline server will provide a native and virtualized environment for configuring, deploying, and running Distributed Pipeline on different hardware platforms.
3. *Data Heterogeneity:* Biological data often include heterogeneous information, such as clinical, genetic, phenotypic, and imaging data. Moreover, these data can be large (often measured in Gigabytes). These two characteristics necessitate care in the design and execution of data processing protocols. Frequently, the processing of heterogeneous data is performed by independent analyses within each data type followed by *ad hoc* strategies for integration, visualization, and interpretation. For instance, neuroimaging genetics studies (Ho et al., 2010a,b) utilize imaging, genetic, and phenotypic data, but most bioinformatics data analysis tools enable processing of only uni-modal spatiotemporal, sequence, or spreadsheet type data. The joint modeling and analysis of such multiform data will significantly increase our ability to discover complex associations, biomarkers, and traits that are currently implicit in the complex genomics data. The Distributed Pipeline study-design mechanism will enable the integration of imaging and meta-data as well as the construction of complete study protocols using the entire data collection. For instance, Distributed Pipeline will enable dynamic decision making, branching, and looping based on the meta-data and on data derived in the analysis protocol itself. Another data-related challenge includes anonymization and/or de-identification of hosted data, to comply with IRB/HIPAA regulations, protect personal information and ensure subject privacy. The

LONI/INI infrastructure provides a two-tier mechanism for data de-identification. First the Imaging Data Archive system ensures that all data (imaging, genetic, demographic) submitted to the database excludes all personal identifiable information (http://www.loni.usc.edu/Software/DiD). Second, the Pipeline environment provides customizable modules for data anonymization, which can be included in the beginning of any graphical processing workflow to ensure the protocol generates intermediate and final results excluding personal information.

4. *Result Reproducibility:* Genomics and informatics protocol dissemination, study replication, and reproducibility of findings have become increasingly important in scientific investigation. Dissemination includes technical publications, distribution of data, URL links, software tools, and execution scripts, as well as screencast, videos, tutorials, and training. Most of these methods for distribution of novel research protocols do not enable outside investigators to independently and efficiently test, validate, or replicate newly proposed techniques. As a result, investigators may frequently reinvent analysis protocols, fail to follow exact procedures, or misinterpret alternative findings. Even when there is a clear description of the scientific model employed in a study (e.g., general linear model), there may be differences in the algorithmic implementation, hardware platform, compiler, environment configuration, or execution-syntax, which can cause differences in the results even using the same input data. The Distributed Pipeline infrastructure will enable flexible and efficient distribution of published (peer-reviewed) workflows, which will facilitate result reproducibility and validation of analysis protocols by the entire user community. Previously developed, validated and published workflows are available online (http://pipeline.loni.usc.edu/explore/library-navigator/).
5. *Steep Learning Curve:* Other informatics challenges include steep learning curves for utilizing general distributed computing environments, and incompatible differences in communication protocols. Currently, significant technical knowledge is required to configure, utilize, and link diverse sequence analysis tools. This task is typically done by developing sophisticated scripts and/or repackaging software resources within specific graphical workflow environments. The Distributed Pipeline computational library will contain a large number of data references and software resources. The included XML data, module, and workflow descriptions will abstract many of the technical details about the standard and advanced features of these resources and promote appropriate access, easy use and efficient modification of the entire compendium of resources available within the Distributed Pipeline library.

Three notable successes include the Biomedical Informatics Research Network (BIRN), the International Neuroinformatics Coordinating Facility (INCF) and the cancer Biomedical Informatics Grid (caBIG). BIRN is a national initiative focused on advancing biomedical research through data sharing and online collaboration. It is funded by the National Institute of General Medicine Sciences (NIGMS), and provides data-sharing infrastructure, software tools, strategies and advisory services—all from a single source (Keator et al., 2008). INCF supports a collaborative neuroinformatics infrastructure and promotes the sharing of data and computing resources to the international research community. INCF is funded by contributions from its member countries, based on gross domestic expenditures on research and development (GERD), www.incf.org. The caBIG program developed and supports access to digital capabilities essential to enhancing researchers' capacity to utilize biomedical information. The initiative aims to disseminate and promote the use of open source standards for data exchange and interoperability in cancer research, develop, maintain, enhance, and share innovative biomedical informatics capabilities, and facilitate the management and analysis of big and heterogeneous cancer research data sets (von Eschenbach and Buetow, 2006).

In this paper, we present the novel infrastructure at the USC Institute for Neuroimaging and Informatics, which is available to the entire computational neuroscience community and addresses many of the current computational neuroscience barriers—lack of integrated storage, hardware, software and processing Big Data infrastructure, limitations of current infrastructure for processing of complex and incomplete data, and the difficulties with resource interoperability.

RESOURCE INFRASTRUCTURE

The INI provides an extensive infrastructure designed and operated to facilitate modern informatics research and support for hundreds of projects including several multi-site national and global efforts. We have redundancies built in to all equipment, and a secure facility to protect equipment and data. The resources described below provide networking, storage and computational capabilities that will ensure a stable, secure and robust environment. It is an unprecedented test bed to create and validate big data solutions. Because these resources have been designed, built and continuously upgraded over the years by our systems administration team, we have the appropriate expertise and operating procedures in place to use these resources to their maximum benefit.

The INI/LONI data center contains a 300 KVa UPS/PDU capable of providing uninterruptible power to mission-critical equipment housed in the room, dual 150 KVa connections to building power, an 800 kW Caterpillar C27 diesel backup generator, three Data Aire computer room air conditioning (CRAC) units, humidity control, and a Cisco fire suppression and preaction system. A sophisticated event notification system is integrated in this space to automatically notify appropriate personnel of any detrimental power and HVAC issues that arise.

DATA CENTER SECURITY

The LONI datacenter is secured by two levels of physical access, to insure HIPAA compliance for data security. The main facility is secured 24/7 with access control devices. Only authorized personnel are allowed in, and guests are permitted only after checking in, and only during business hours. The datacenter itself is additionally secured by a second layer of proximity card access. Only authorized staffs are permitted to enter the datacenter facility. Individual racks containing HIPAA data are secured by lock and key to prevent cross access.

COMPUTATIONAL AND STORAGE RESOURCES

Rapid advancements in imaging and genetics technology have provided researchers with the ability to produce very high-resolution, time-varying, multidimensional data sets of the brain. The complexity of the new data, however, requires immense computing capabilities. The compute infrastructure within the datacenter boasts 3328 cores and 26 Tb of aggregate memory space, **Figure 1**. This highly available, redundant system is designed for demanding big data applications. Blades in the Cisco UCS environment are easy to replace. A failing blade sends an alert to Cisco where a replacement ticket is generated automatically. Upon arrival, the new blade can go from the shipping box to being fully provisioned and in production in as little as 5 min. Institutions and scientists worldwide rely on the LONI's resources to conduct research. LONI is architected using a fault-tolerant, high-availability systems design to ensure 24/7 functionality. The primary storage cluster is 23 Isilon nodes with 2.4 usable petabytes of highly available, high performance storage. Data in these clusters moves exclusively over 10 g links excepting node to node communication in the Isilon cluster which is handled by QDR Infiniband, providing 40 gigabit bidirectional throughput on each of the Isilon cluster's 46 links. Fault tolerance is as important as speed in the design of this datacenter. The Isilon storage cluster can gracefully lose multiple nodes simultaneously without noticeably affecting throughput or introducing errors.

External services are load balanced across four F5 BIG-IP 2200S load balancers. The F5 load balancers provide balancing services for web sites, applications, as well as ICSA-certified firewall services. The INI core network is entirely Cisco Nexus hardware. Each of the two Cisco Nexus 5596 s supports 1.92 Tb per second of throughput. Immediately adjacent to this machine room is a user space with twelve individual stations separated by office partitions. These workspaces are manned by staff who constantly monitor the health of the data center as well as plan for future improvements. Each space is also equipped with a networked workstation for image processing, visualization and statistical analysis.

NETWORK RESOURCES

Service continuity, deterministic performance and security were fundamental objectives that governed the design of LONI's network infrastructure. The laboratory intranet is architected using separate edge, core and distribution layers, with redundant switches in the edge and core for high availability, and with Open Shortest Path First (OSPF) layer 3 routing, instead of a traditional flat layer 2 design, to leverage the fault tolerance offered by packet routing and to minimize network chatter. While ground network connectivity is entirely Gigabit, server data connectivity is nearly all 10 Gb fiber and Twinax connected to a core of 2 Cisco Nexus 5596 switches, 10 Cisco Nexus 6628 switches, and 6 Cisco Nexus 2248 fabric extenders. For Internet access, INI is connected to the

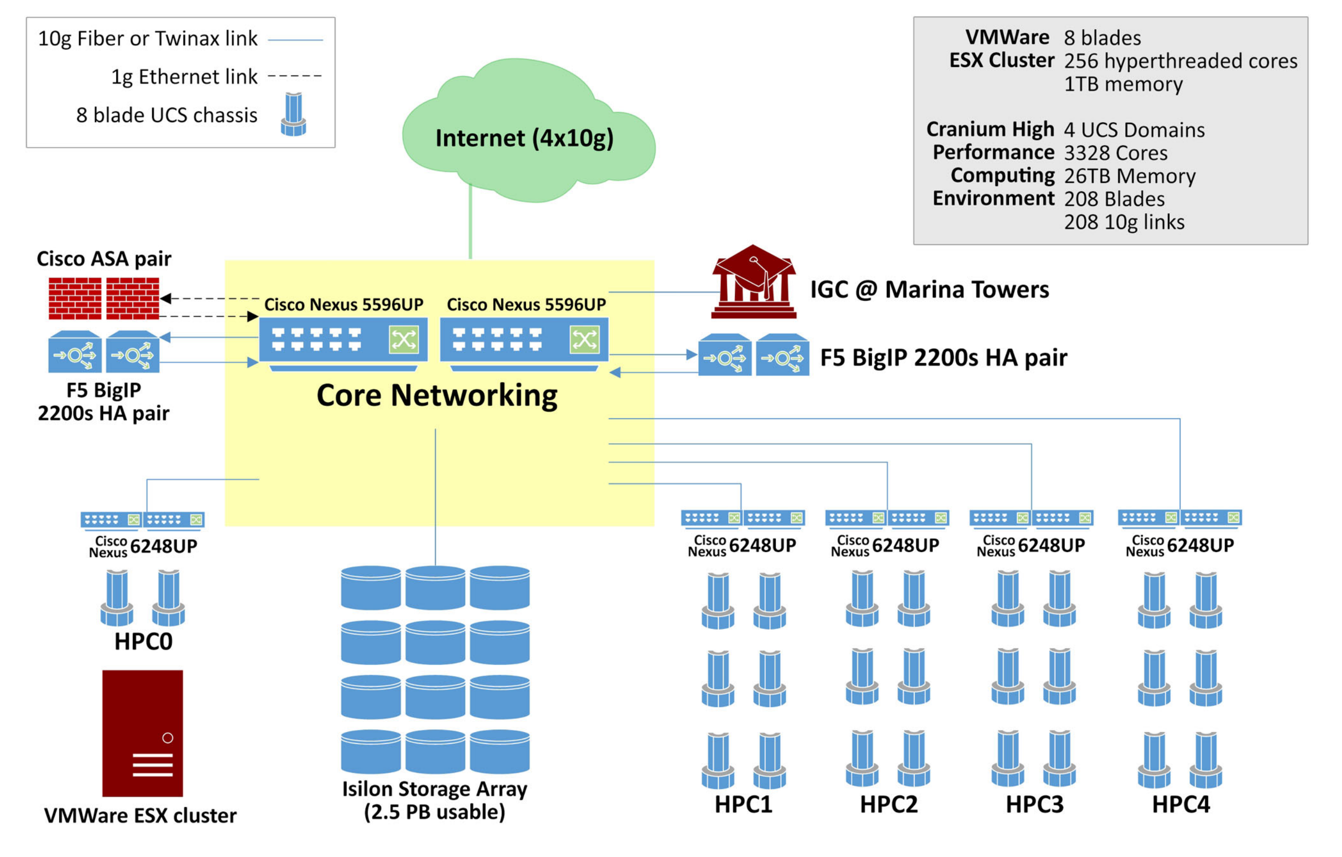


FIGURE 1 | LONI/INI network infrastructure and supercomputing environnement.

vBNS of Internet2 via quad fiber optic Gigabit lines using different route paths to ensure that the facility's external connectivity will be maintained in the case of a single path failure.

The facility has two Cisco Adaptive Security Appliances providing network security and deep packet inspections. LONI has also implemented virtual private network (VPN) services using SSLVPN and IPsec services to facilitate access to internal resources by authorized users. A VPN connection establishes an encrypted tunnel over the Internet between client and server, ensuring that communications over the Web are secure. Furthermore, the laboratory has an extensive library of communications software for transmitting data and for recording transaction logs. The library includes software for monitoring network processes, automatically warning system operators of potential problems, restarting processes that have failed, or migrating network services to an available server. For instance, the laboratory has configured multiple web servers with Linux Virtual Server (LVS) software for high-availability web, application and database service provisioning as well as load balancing. A round-robin balancing algorithm is currently used such that if the processing load on one server is heavy, incoming requests, be it HTTP, JSP or MySQL, are forwarded to the next available server by the LVS software layer. Listeners on one virtual server monitor the status and responsiveness of the others. If a failure is detected, an available server is elected as master and it assumes control and request forwarding for the entire LVS environment.

VIRTUALIZED RESOURCES

Due to the rate that new servers need to be provisioned for scientific research, INI deploys a sophisticated high availability virtualized environment. This environment allows INI systems administrators to deploy new compute resources (virtual machines or VM's) in a matter of minutes rather than hours or days. Furthermore, once deployed, these virtualized resources can float uninhibitedly between all the physical servers within the cluster. This is advantageous because the virtualization cluster can intelligently balance virtual machines amongst all the physical servers, which permits resource failover if a virtual machine becomes I/O starved or a physical server becomes unavailable. The net benefit for LONI is more software resources are being efficiently deployed on a smaller hardware footprint, which results in a savings in hardware purchases, rack space and heat expulsion.

The software powering LONI virtualized environment is VMware's ESX 5. The ESX 5 is deployed on eight Cisco UCS B200 M3 servers, each with sixteen 2.6/3.3 GHz CPU cores and 128 Gb of DDR3 RAM. These eight servers reside within a Cisco UCS 5108 blade chassis with dual 8 × 10 Gb mezzanine cards providing a total of 160 Gb of available external bandwidth. Storage for the virtualization cluster is housed on the 23 nodes of Isilon storage. The primary bottleneck for the majority of virtualization solutions is disk I/O and the Isilon cluster more than meets the demands of creating a highly available virtualized infrastructure whose capabilities and efficiency meet or greatly exceed those of a physical infrastructure. A single six rack unit (6RU), eight blade chassis can easily replicate the resources of a 600+ server physical infrastructure when paired with the appropriate storage solution such as the INI Isilon storage cluster.

WORKFLOW PROCESSING

To facilitate the submission and execution of compute jobs in this compute environment, various batch-queuing systems such as SGE (https://arc.liv.ac.uk/trac/SGE) can be used to virtualize the resources above into a compute service. A grid layer sits atop the compute resources and submits jobs to available resources according to user-defined criteria such as CPU type, processor count, memory requirements, etc. The laboratory has successfully integrated the latest version of the LONI Pipeline (http://pipeline.loni.usc.edu) with SGE using DRMAA and JGDI interface bindings (Dinov et al., 2009, 2010; Torri et al., 2012). The bindings allow jobs to be submitted natively from the LONI Pipeline to the grid without the need for external scripts. Furthermore, the LONI Pipeline can directly control the grid with those interfaces, significantly increasing the operating environment's versatility and efficacy, and improving overall end-user experience. **Figure 2** illustrates the latest version of the pipeline client software.

The data center will be approximately 3000 square feet and is being designed using cutting edge high density cooling solutions and high density bladed compute solutions. A total of 48 racks will be installed and dedicated to research use. Of the 48, 10 racks will be reserved for core services. The core services are on separate, dedicated, redundant power to ensure continuous operation. The current design of the data center includes a Powerware 9395 UPS system providing two 750 kW/825 kVA UPSs in a N+1 configuration for non-core racks and two 225 kW/250 kVA in a 2N configuration for core services racks. The UPS sends conditioned power to 300 kVA Power Distribution Units (PDUs) located inside the data center. The PDUs feed 400 A rated Track Power Busways mounted above rows of racks providing an "A" bus and a "B" bus for flexible overhead power distribution to the racks. The design calls for the use of VRLA batteries with 9 min of battery run time for the core services UPS and 6 min of battery run time for the non-core UPS (note that the generator requires less than 2 min of battery run time in order to fully take over the load in the event of an outage). A 750 kW/938 kVA diesel emergency generator located in a weatherproof sound attenuated enclosure adjacent to the building will provide at least 8 h of operation before needing to be refueled.

The Cisco UCS blade solution described above allows LONI to run the services of a much larger physical infrastructure in a much smaller footprint without sacrificing availability or flexibility. Each Cisco chassis hosts 8 server blades and has 160 Gb of external bandwidth available per chassis. Each of the 48 racks can hold up to 6 chassis plus requisite networking equipment (4 fabric extenders). Thus, the new data center has adequate rack space to accommodate this project.

In addition to a new data center, the INI infrastructure will house a 50-seat high definition theater—the Data Immersive Visualization Environment (DIVE). The prominent feature of the DIVE is a large curved display that can present highly detailed images, video, interactive graphics, and rich media generated by specialized research data. The DIVE display will feature a dominant image area, with consistent brightness across the entire display surface, high contrast, and 150° horizontal viewing angle. The display resolution target is 4 k Ultra HD, 3840 × 2160 (8.3 megapixels), in a 16:9 aspect ratio. Due to the ceiling height

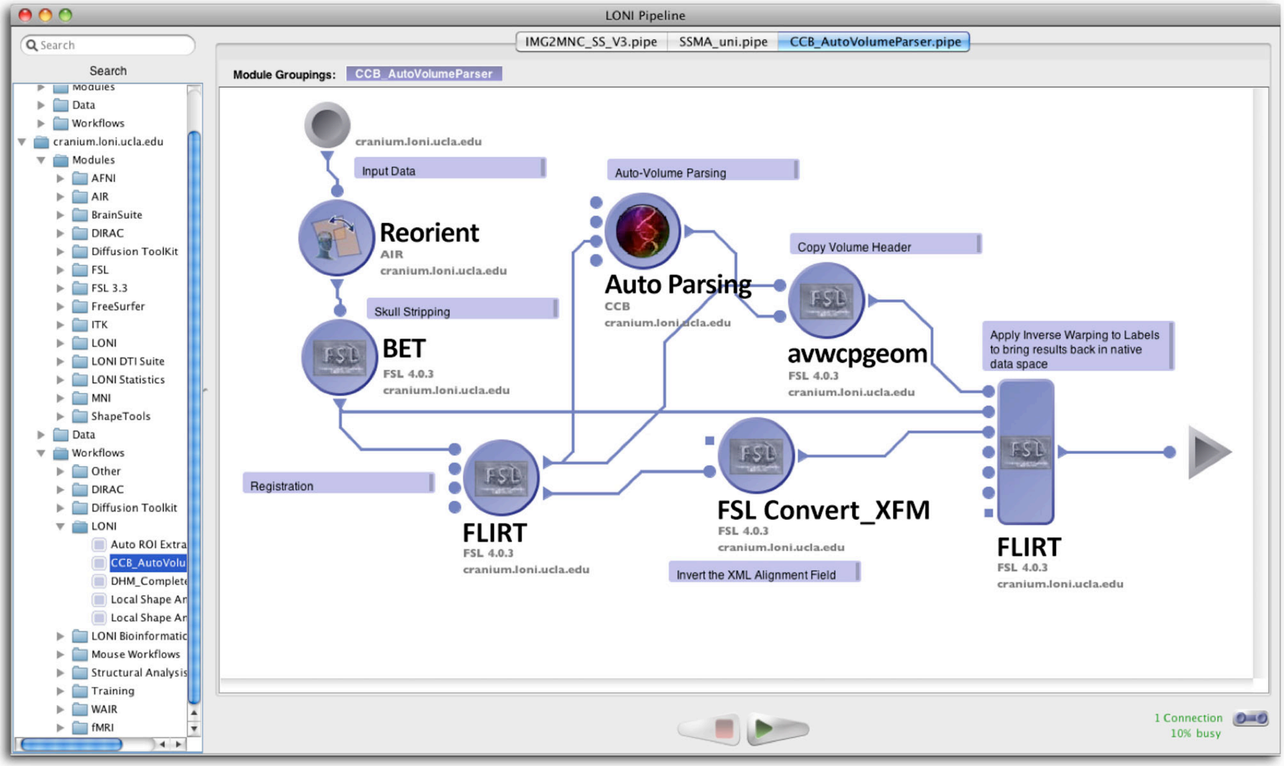


FIGURE 2 | The client interface to the LONI Pipeline execution environment.

requirements, the DIVE will require two floors of the building. The DIVE is designed to facilitate research communication, dissemination, training, and high levels of interaction.

EXEMPLARY STUDIES

ALZHEIMER'S DISEASE IMAGING-GENETICS STUDY

Using subjects over the age of 65 from the Alzheimer's Disease Neuroimaging Initiative (ADNI) archive, http://adni.loni.usc.edu (Weiner et al., 2012), we investigated cognitive impairment using neuroimaging and genetic biomarkers. Querying the ADNI database, we selected 808 participants including 200 Alzheimer's Disease (AD) patients (108 males and 92 females), 383 mild cognitive impairment (MCI) subjects (246 males and 137 females), and 225 asymptomatic normal control (NC) volunteers (116 males and 109 females). After downloading the individual ADNI imaging data we carried standard quality control genetic analysis, using PLINK version 1.09, (Purcell et al., 2007). All data analytics were performed using the LONI Pipeline environment (Dinov et al., 2010; Torri et al., 2012). The global shape analysis protocol provides a set of 20 derived neuroimaging markers ($P < 0.0001$, between group ANOVA), which are studied in the context of the 20 most significant single nucleotide polymorphisms (SNPs), chosen by Manhattan plot, associated with the AD, MCI, and NC cohorts, as subject phenotypes. The structural ADNI data (1.5T MRI) were parcellated using BrainParser (Tu et al., 2008). The complete data analysis protocol and some of the intermediate results are shown on **Figure 3**.

This large scale study identified that neuroimaging phenotypes were significantly associated with the progression of dementia from NC to MCI and ultimately to AD. Our results pooling MCI and AD subjects together ($N_1 = 583$) compared to NC subjects ($N_2 = 225$) indicates significant association between 20 SNPs and 2 neuroimaging phenotypes as shown in the heatmap plot, **Figure 4**. The data analytics presented in this case study demand significant data storage, processing power and bandwidth capabilities to accomplish the end-to-end data processing, analysis and visualization. In this study the protocol includes about 100 independent processing steps and the analysis tool 2 days on a 1200 compute node shared cluster.

PARKINSON DISEASE ANALYTICS

There is some clinical evidence that the different subtypes of Parkinson's disease (PD) may follow different clinical courses. Tremor-dominant cohorts show a slower progress of the disease and less cognitive decline than akinetic rigid group (Kang et al., 2013). The clinical subtypes probably are in concordance with differences in brain biochemical abnormalities. In this example, using the Parkinson Progression Marker Initiative (PPMI) brain data (Marek et al., 2011), we analyze structural brain changes in Parkinson's disease relative to their relationship with subtypes of Parkinson's disease. Specifically, the goal was to utilize the INI/LONI computational infrastructure to study interrelations between subtypes and biomedical imaging features in 150 PPMI subjects. This analysis protocol includes automatic generation of

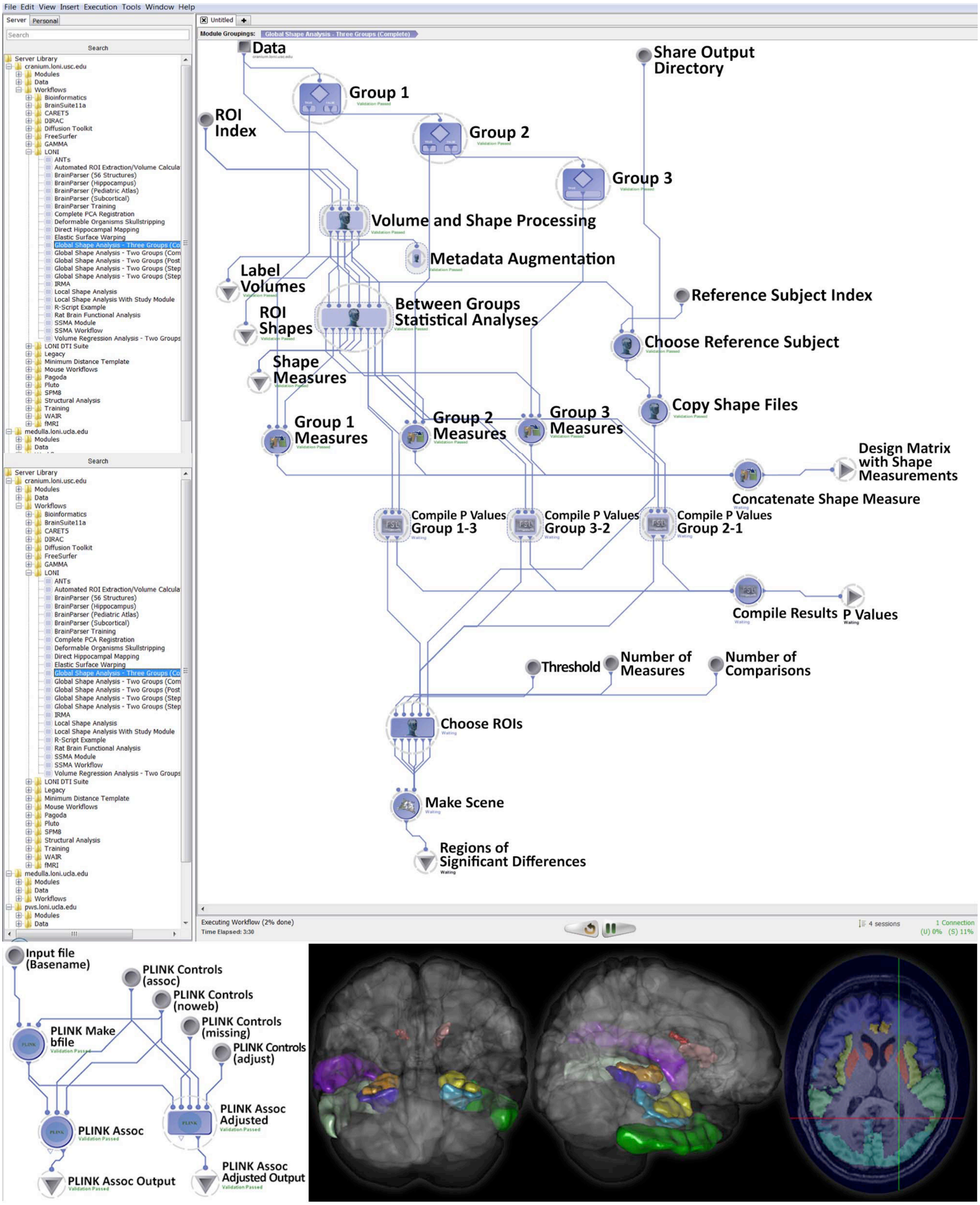


FIGURE 3 | Global shape analysis (GSA) protocol extracting neuroimaging biomarkers for each of the 3 cohorts (top), genetic phenotyping (bottom left), and examples of intermediate derived neuroimaging biometrics (bottom right).

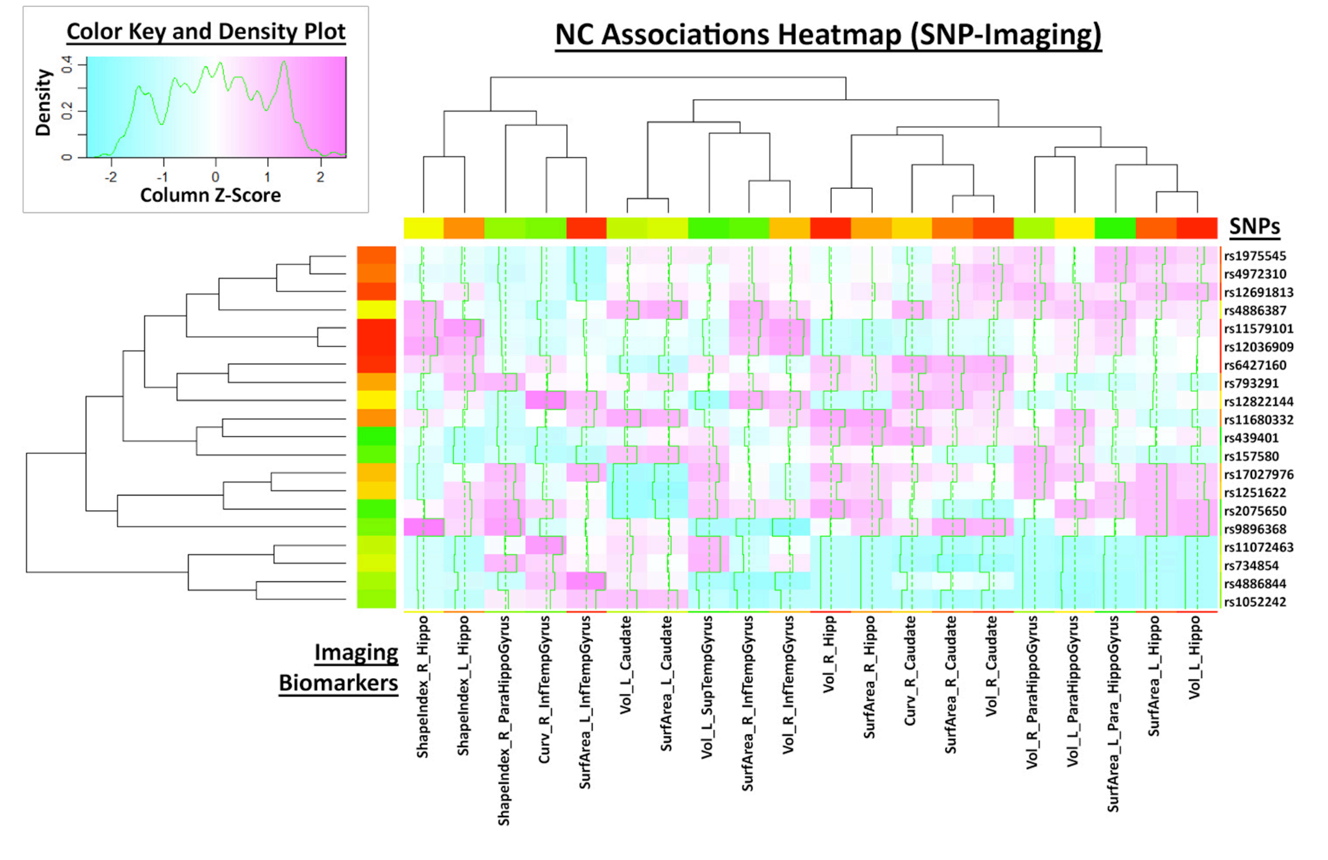


FIGURE 4 | Heatmap plot of 20 neuroimaging derivative measures associated with the subject phenotype (columns) and the SNP genotypes.

56 regions of interest (ROIs) for each subject and computing various volume-based and shape-based measures for each region of interest (ROI), e.g., volume, dice coefficient, overlap measure, mean curvature, surface area, mean fractal dimension, shape-index, curvedness) (Dinov et al., 2010). **Figure 5** illustrates a high-level view of the morphometric analysis of the data (left) and an example of an automatically generated gray matter thickness map for one of the processed cases. This workflow completed on the INI cluster in 2 days, competing with thousands of other processes that are run in parallel and submitted by different users.

Although both of these examples demonstrate a small fraction of the available processing modules and end-to-end computational workflow solutions, the Pipeline environment includes a much larger library of resources for image processing (Dinov et al., 2009), shape analysis (Dinov et al., 2010), next generation sequence analysis (Torri et al., 2012), and bioinformatics. These examples were chosen as they indicate demand for significant computational power to process hundreds of cases in parallel, the ability to handle high-throughput data transfer (near real time) with access to external databases, and the software necessary to pre-process, model, integrate, and visualize large multivariate datasets.

DISCUSSION

The neuroscience of the Twentieth Century was built upon the Popperian ideal of forming questions suitable as empirical hypotheses to be tested using experimentally derived data. Yet, with modern neuroimaging and genomics technologies, we are now able to gather more data per experiment that was gathered in perhaps years of collection 20 years ago. While the philosophy of science ideal based on hypothesis testing has by no means been surpassed, it is clear that the data being obtained offers greater information beyond the hypotheses under test which, indeed, offers more opportunity to explore larger data spaces and therefore form new testable lines for scientific investigation. Thus, in as much as the question itself is the driver of scientific progress, the data being obtained provides the chance to identify new questions worthy of our attention. For which, we will need to gather still more data.

As large quantities of data are gathered for any particular experiment, their accumulation into local databases and publicly available archives (e.g., the LONI Image and Data Archive; http://ida.loni.usc.edu) there is an increasing need for large-scale computational resources such as those discussed above. Be these resources local or remote ("in the cloud"), their availability helps to expedite data analysis, synthesis, its mining, and summarization such that old questions can be readily addressed and new questions can be formulated. With the increases in data size come increased needs to process data faster. The LONI/INI computational systems are one such example of pushing processing capability to the forefront of neuroimaging and genomic analytics. Other resources include Amazon, Microsoft, and Google services

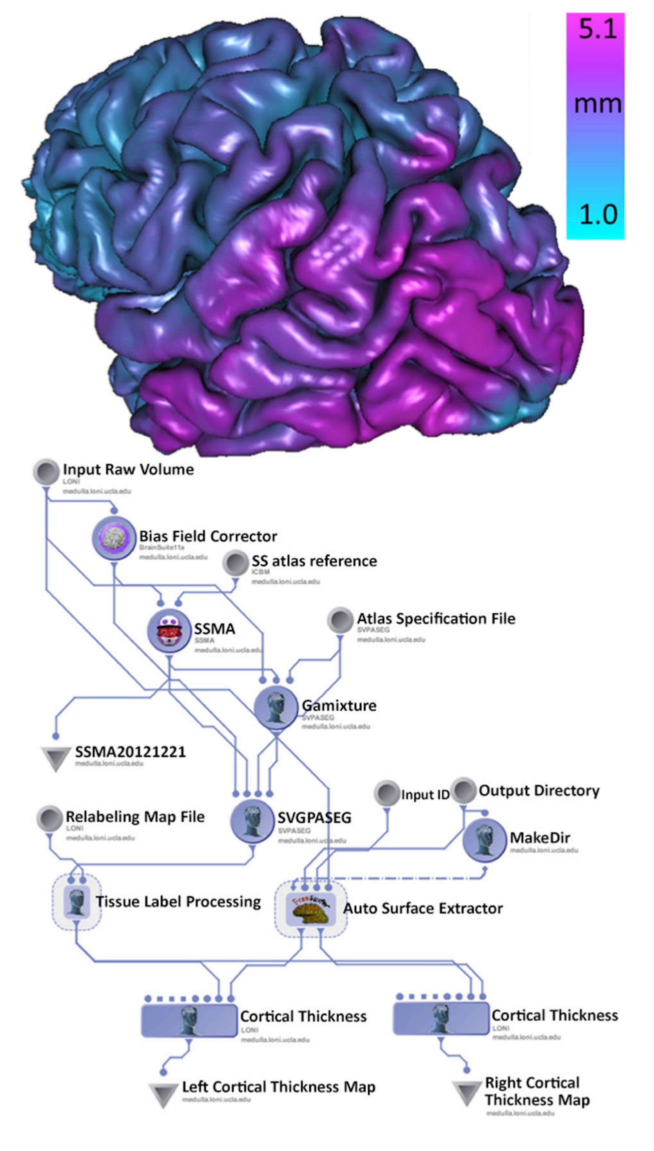


FIGURE 5 | Pipeline workflow protocol for automated extraction of imaging biomarkers and association of imaging and phenotypic PPMI data (left), and a 3D rendering of the cortical surface, colored by the gray matter thickness map, for one individual (right).

which, for a fee, users can provision data storage and multiprocessor virtual systems upon which to configure and perform neuroimaging or genetics analyses. Irrespective of the form the computational infrastructure takes, there is little question that such services are a necessary element for Twenty-first Century biomedical science where data is king.

Data management including archival, query, retrieval, aggregation, and fusion are enabled via the LONI Pipeline Environment. For example, the initial data-sources within each workflow can pull data from different servers, aggregate it into the computational workflow, jointly process it and save intermediate and final results in different locations. As there is a growing array of publicly available data sets, this functionality is critical for large-scale collaborative studies requiring significant sample-sizes to identify associations and relations for (marginal) effect-sizes. Pipeline cloud-based data sources (inputs) and sinks (outputs) are similar to regular data sources and sinks, except that data are stored in the cloud. The Pipeline takes care of the data transfer between the cloud vendor and the compute nodes. Currently supported cloud source vendors include Amazon S3 and Dropbox (http://pipeline.loni.usc.edu/learn/user-guide/building-a-workflow/#Cloud%20sources%20and%20sinks). In addition, users can set up instances replicating the entire Pipeline infrastructure on Amazon EC2 (http://pipeline.loni.usc.edu/products-services/pipeline-server-on-ec2/).

The INI/LONI infrastructure has been specifically designed to meet the big data storage and processing challenges as evident from large-scale, multi-site neuroimaging initiatives such as ADNI, the Autism Centers of Excellence (ACE), PPMI, the Human Connectome Project (Toga et al., 2012), and others. With new NIH programs for brain research on the horizon, the computational systems and processing capabilities described here will find immediate application for the archiving, processing, and mining of vast quantities of neuroscience data from healthy as well as diseased subjects. There are several alternative Cloud-based computational neuroscience resources with similar goals and infrastructure. For example, the Neuroscience Gateway (www.nsgportal.org) portal is supported by the Extreme Science and Engineering Discovery Environment (XSEDE) Resource Allocation Committee and provides High Performance Computing resources for the neuroscience community. The Neuroimaging Tools and Resources Clearinghouse (NITRC) Amazon EC2 Computational Environment is a virtual computing platform configured with many neuroimaging data analysis applications (https://aws.amazon.com/marketplace/pp/B00AW0MBLO?sr=0-2). The INI/LONI infrastructure does have its limitations. System bottlenecks include potential for large number of simultaneous users (hundreds), or a few heavy users (e.g., a dozen users with complex protocols involving tens of thousands of jobs managed in parallel), can significantly impact the performance of the back-end Pipeline server and NFS manager. Data I/O access could be affected when managing a huge number of simultaneous read-write requests, including handling intermediate results. Upgrading the infrastructure (e.g., hardware expansions, system updates, software upgrades) require a significant concerted effort.

INI/LONI welcomes new ideas from the entire computational community and constantly promotes new collaborations with outside investigators. The LONI/INI infrastructure is freely available to the entire community (registration and accounts are required). There is a variety of data, modeling, computational, scientific or translational-research collaborations we support, which can be initiated by completing one of the online web-forms (http://resource.loni.usc.edu/collaboration/collaborator-application/). This manuscript attempts to demonstrate how the entire biomedical community can utilize the LONI resources, as well as demonstrate the design-challenges, capabilities, and maintenance of such integrated data, software and hardware architectures, which may be valuable to others interested in building similar, alternative or federated computational frameworks.

ACKNOWLEDGMENTS

This work was partially supported by National Institutes of Health grants NIA P 50 AG16570, NIBIB EB01651, NLM LM05639, NIMH R01 MH071940, NIBIB P41EB015922, U24-RR025736, U24-RR021992, U24 GM104203. This work was also supported in part by the Global Alzheimer's Association Interactive Network (GAAIN) initiative of the Alzheimer's Association. The authors are also indebted to the faculty, staff, and students in the Laboratory of Neuro Imaging (LONI) for their support and dedication. Some of the data processed in this study was partly funded by the Parkinson's Progression Markers Initiative (PPMI) and Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Alzheimer's Association; Alzheimer's Drug Discovery Foundation; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; GE Healthcare; Innogenetics, N.V.; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of California, Los Angeles. The Parkinson's Progression Markers Initiative (PPMI) is a public-private partnership—is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners, including Abbott, Avid, BioGenIdec, COVANCE, Elan, GE Healthcare, Genentech, GlaxoSmithKline, Eli Lilly, Merck, Pfizer, Roche, UCB (http://www.ppmi-info.org/about-ppmi/who-we-are/study-sponsors/). The PPMI database is available online (www.ppmi-info.org/data), and up-to-date information on the PPMI study is online at www.ppmi-info.org.

REFERENCES

Barker, A., and Van Hemert, J. (2008). Scientific workflow: a survey and research directions. *Lect. Notes Comput. Sci.* 4967, 746–753. doi: 10.1007/978-3-540-68111-3_78

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., et al. (2009). NCBI GEO: archive for high-throughput functional genomic data. *Nucl. Acids Res.* 37(suppl. 1), D885–D890. doi: 10.1093/nar/gkn764

Blokland, G. A. M., McMahon, K. L., Hoffman, J., Zhu, G., Meredith, M., Martin, N. G., et al. (2008). Quantifying the heritability of task-related brain activation and performance during the N-back working memory task: a twin fMRI study. *Biol. Psychol.* 79, 70–79. doi: 10.1016/j.biopsycho.2008.03.006

Dinov, I. D., Petrosyan, P., Liu, Z., Eggert, P., Zamanyan, A., Torri, F., et al. (2013). The perfect neuroimaging-genetics-computation storm: collision of petabytes of data, millions of hardware devices and thousands of software tools. *Brain Imaging Behav.* 1–12. doi: 10.1007/s11682-013-9248-x

Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., et al. (2010). Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline. *PLoS ONE* 5:e13070. doi: 10.1371/journal.pone.0013070

Dinov, I., Rubin, D., Lorensen, W., Dugan, J., Ma, J., Murphy, S., et al. (2008). iTools: a framework for classification, categorization and integration of computational biology resources. *PLoS ONE* 3:e2265. doi: 10.1371/journal.pone.0002265

Dinov, I., Van Horn, J., Lozev, K., Magsipoc, R., Petrosyan, P., Liu, Z., et al. (2009). Efficient, distributed and interactive neuroimaging data analysis using the LONI pipeline. *Front. Neuroinform.* 3, 1–10. doi: 10.3389/neuro.11.022.2009

Evans, A. C. (2006). The NIH MRI study of normal brain development. *Neuroimage* 30, 184–202. doi: 10.1016/j.neuroimage.2005.09.068

Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* 11, R86. doi: 10.1186/gb-2010-11-8-r86

Gousias, I. S., Rueckert, D., Heckemann, R. A., Dyet, L. E., Boardman, J. P., Edwards, A. D., et al. (2008). Automatic segmentation of brain MRIs of 2-year-olds into 83 regions of interest. *Neuroimage* 40, 672–684. doi: 10.1016/j.neuroimage.2007.11.034

Ho, A. J, Raji, C. A., Becker, J. T., Lopez, O. L., Kuller, L. H., Hua, X., et al. (2010a). Obesity is linked with lower brain volume in 700 AD and MCI patients. *Neurobiol. Aging* 31, 1326–1339. doi: 10.1016/j.neurobiolaging.2010.04.006

Ho, A. J., Stein, J. L., Hua, X., Lee, S., Hibar, D. P., Leow, A. D., et al. (2010b). A commonly carried allele of the obesity-related FTO gene is associated with reduced brain volume in the healthy elderly. *Proc. Natl. Acad. Sci.* 107, 8404–8409. doi: 10.1073/pnas.0910878107

Kandel, E. R., Markram, H., Matthews, P. M., Yuste, R., and Koch, C. (2013). Neuroscience thinks big (and collaboratively). *Nat. Rev. Neurosci.* 14, 659–664. doi: 10.1038/nrn3578

Kang, J.-H., Irwin, D. J., Chen-Plotkin, A. S., Siderowf, A., Caspell, C., Coffey, C. S., et al. (2013). Association of cerebrospinal fluid β-Amyloid 1-42, T-tau, P-tau181, and α-Synuclein levels with clinical features of Drug-naive patients with early Parkinson DiseaseAβ1-42, T-tau, P-tau181, α-Synuclein, and PDAβ1-42, T-tau, P-tau181, α-Synuclein, and PD. *JAMA Neurol.* 70, 1277–1287. doi: 10.1001/jamaneurol.2013.3861

Keator, D. B., Grethe, J. S., Marcus, D., Ozyurt, B., Gadde, S., Murphy, S., et al. (2008). A national human neuroimaging collaboratory enabled by the Biomedical Informatics Research Network (BIRN). *Inf. Technol. Biomed. IEEE Trans.* 12, 162–172. doi: 10.1109/TITB.2008.917893

Kubica, S., Robey, T., and Moorman, C. (1998). Data parallel programming with the Khoros Data Services Library. *Lect. Notes Comput. Sci.* 1388, 963–973. doi: 10.1007/3-540-64359-1_762

Liu, M., Zhang, D., and Shen, D. (2012). Ensemble sparse classification of Alzheimer's disease. *Neuroimage* 60, 1106–1116. doi: 10.1016/j.neuroimage.2012.01.055

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., et al. (2006). Scientific workflow management and the Kepler system. *Concurrency Comput. Pract. Exper.* 18, 1039–1065. doi: 10.1002/cpe.994

Marek, K., Jennings, D., Lasch, S., Siderowf, A., Tanner, C., Simuni, T., et al. (2011). The Parkinson Progression Marker Initiative (PPMI). *Prog. Neurobiol.* 95, 629–635. doi: 10.1016/j.pneurobio.2011.09.005

Mazziotta, J. C., Toga, A. W., Evans, A., Fox, P., and Lancaster, J. (1995). A probabilistic atlas of the human brain: theory and rationale for its development. The International Consortium for Brain Mapping (ICBM). *Neuroimage* 2, 89–101. doi: 10.1006/nimg.1995.1012

Mueller, S. G., Weiner, M. W., Thal, L. J., Petersen, R. C., Jack, C. R., Jagust, W., et al. (2005). Ways toward an early diagnosis in Alzheimer's disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI). *Alzheimer's Dement.* 1, 55–66. doi: 10.1016/j.jalz.2005.06.003

Novak, N. M., Stein, J. L., Medland, S. E., Hibar, D. P., Thompson, P. M., and Toga, A. W. (2012). EnigmaVis: online interactive visualization of genome-wide association studies of the Enhancing NeuroImaging Genetics through Meta-Analysis (ENIGMA) consortium. *Twin Res. Hum. Genet.* 15, 414. doi: 10.1017/thg.2012.17

Oinn, T., Greenwood, M., Addis, M., Alpdemir, M. N., Ferris, J., Glover, K., et al. (2005). Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency Comput. Pract. Exp.* 18, 1067–1100. doi: 10.1002/cpe.993

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A. R., Bender, D., et al. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* 81, 559–575. doi: 10.1086/519795

Rex, D. E., Ma, J. Q., and Toga, A. W. (2003). The LONI pipeline processing environment. *Neuroimage* 19, 1033–1048. doi: 10.1016/S1053-8119(03)00185-X

Simmhan, Y. L., Plale, B., and Gannon, D. (2008). Karma2: provenance management for data driven workflows. *Int. J. Web Serv. Res.* 5, 1–22. doi: 10.4018/jwsr.2008040101

Sloutsky, R., Jimenez, N., Swamidass, S. J., and Naegle, K. M. (2013). Accounting for noise when clustering biological data. *Brief. Bioinform.* 14, 423–436. doi: 10.1093/bib/bbs057

Toga, A. W., Clark, K. A., Thompson, P. M., Shattuck, D. W., and Van Horn, J. D. (2012). Mapping the human connectome. *Neurosurgery* 71, 1–5. doi: 10.1227/NEU.0b013e318258e9ff

Torri, F., Dinov, I. D., Zamanyan, A., Hobel, S., Genco, A., Petrosyan, P., et al. (2012). Next generation sequence analysis and computational genomics using graphical pipeline workflows. *Genes* 3, 545–575. doi: 10.3390/genes3030545

Tu, Z., Narr, K. L., Dinov, I., Dollar, P., Thompson, P. M., and Toga, A. W. (2008). Brain anatomical structure segmentation by hybrid discriminative/generative models. *IEEE Trans. Med. Imaging* 27, 495–508. doi: 10.1109/TMI.2007.908121

Van Horn, J. D., and Toga, A. W. (2013). Human neuroimaging as a “Big Data” science. *Brain Imaging Behav.* 1–9. doi: 10.1007/s11682-013-9255-y

von Eschenbach, A. C., and Buetow, K. (2006). Cancer informatics vision: caBIG™. *Cancer Inform.* 2, 22–24. Available online at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2675495/

Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., et al. (2012). The Alzheimer’s Disease Neuroimaging Initiative: a review of papers published since its inception. *Alzheimer’s Dement.* 8, S1–S68. doi: 10.1016/j.jalz.2011.09.172

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 07 January 2014; accepted: 27 March 2014; published online: 23 April 2014.
Citation: Dinov ID, Petrosyan P, Liu Z, Eggert P, Hobel S, Vespa P, Woo Moon S, Van Horn JD, Franco J and Toga AW (2014) High-throughput neuroimaging-genetics computational infrastructure. Front. Neuroinform. ***8****:41. doi: 10.3389/fninf.2014.00041*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 14 March 2014
doi: 10.3389/fninf.2014.00024

BROCCOLI: Software for fast fMRI analysis on many-core CPUs and GPUs

Anders Eklund[1]*, Paul Dufort[2], Mattias Villani[3] and Stephen LaConte[1,4]

[1] Virginia Tech Carilion Research Institute, Virginia Tech, Roanoke, VA, USA
[2] Department of Medical Imaging, University of Toronto, Toronto, ON, Canada
[3] Division of Statistics, Department of Computer and Information Science, Linköping University, Linköping, Sweden
[4] School of Biomedical Engineering and Sciences, Virginia Tech-Wake Forest University, Blacksburg, VA, USA

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Roy Ben-Shalom, Bar-Ilan University, Israel
Moises Hernandez Fernandez, The Oxford Centre for Functional MRI of the Brain, UK

Correspondence:
Anders Eklund, Virginia Tech Carilion Research Institute, Virginia Tech,
2 Riverside Circle, Roanoke,
240 16 VA, USA
e-mail: andek034@gmail.com

Analysis of functional magnetic resonance imaging (fMRI) data is becoming ever more computationally demanding as temporal and spatial resolutions improve, and large, publicly available data sets proliferate. Moreover, methodological improvements in the neuroimaging pipeline, such as non-linear spatial normalization, non-parametric permutation tests and Bayesian Markov Chain Monte Carlo approaches, can dramatically increase the computational burden. Despite these challenges, there do not yet exist any fMRI software packages which leverage inexpensive and powerful graphics processing units (GPUs) to perform these analyses. Here, we therefore present BROCCOLI, a free software package written in OpenCL (Open Computing Language) that can be used for parallel analysis of fMRI data on a large variety of hardware configurations. BROCCOLI has, for example, been tested with an Intel CPU, an Nvidia GPU, and an AMD GPU. These tests show that parallel processing of fMRI data can lead to significantly faster analysis pipelines. This speedup can be achieved on relatively standard hardware, but further, dramatic speed improvements require only a modest investment in GPU hardware. BROCCOLI (running on a GPU) can perform non-linear spatial normalization to a 1 mm^3 brain template in 4–6 s, and run a second level permutation test with 10,000 permutations in about a minute. These non-parametric tests are generally more robust than their parametric counterparts, and can also enable more sophisticated analyses by estimating complicated null distributions. Additionally, BROCCOLI includes support for Bayesian first-level fMRI analysis using a Gibbs sampler. The new software is freely available under GNU GPL3 and can be downloaded from github (https://github.com/wanderine/BROCCOLI/).

Keywords: Neuroimaging, fMRI, Spatial normalization, GPU, CUDA, OpenCL, Image registration, Permutation test

1. INTRODUCTION

Functional magnetic resonance imaging (fMRI) has become the de facto standard methodology in contemporary efforts to image the functioning of the human brain in both health and disease. Nonetheless, fMRI-based research arguably lags behind in its adoption of recent advances in computer hardware, despite several recent trends that have underlined the need for greater computational resources. First, the temporal and the spatial resolution of fMRI data continues to improve with stronger magnetic fields and more advanced scanning protocols (Moeller et al., 2010; Feinberg and Yacoub, 2012), leading to the production of significantly larger datasets. Second, fMRI studies are trending toward larger numbers of subjects to increase their statistical power (Eklund et al., 2012a; Thyreau et al., 2012; Button et al., 2013) sometimes aided by a proliferation of data sharing initiatives (Biswal et al., 2010; Poldrack et al., 2013)[1, 2] that provide open access to large amounts of data. The human connectome project (van Essen et al., 2013)[3], for example, shares high resolution data from a large number of subjects (the goal is 1200), and a single resting state scan results in a dataset of the size 104 × 90 × 72 × 1200. Third, non-parametric methods based on permutation and Bayesian Markov Chain Monte Carlo (MCMC) methods are more frequently being used to improve neuroimaging statistics (da Silva, 2011; Eklund et al., 2012a, 2013b), but suffer from long processing times compared to conventional parametric methods. Some progress toward parallelization has been made in each of the three major packages commonly used in fMRI-based research (SPM, FSL, and AFNI). For example, AFNI has direct support for running some functions in parallel on several CPU cores, using the open multi-processing (OpenMP) library; FSL can take advantage of several computers or CPU cores, by installing packages like Condor or GridEngine, and has recently added graphics processing unit (GPU) support for MCMC based diffusion tensor analysis (Hernandez et al., 2013); and Huang et al. (2011) recently proposed to accelerate image

[1] http://fcon_1000.projects.nitrc.org/fcpClassic/FcpTable.html
[2] https://openfmri.org/
[3] http://www.humanconnectome.org/

registration in SPM by using a GPU. Moreover, a number of prominent projects are underway to enable big data approaches to functional neuroimaging at large supercomputing centers [e.g., (Lavoie-Courchesne et al., 2012)]. At this stage, however, these approaches still require a significant investment of time and effort by expert technical staff, and thus remain inaccessible to the majority of investigators. Thus, despite efforts by existing analysis packages, we feel that the community could benefit from a more comprehensive focus on parallel computation. Further, being relatively new, GPUs offer some unique challenges as well as promising potential benefits.

Since the introduction of the CUDA programming language in 2007, general purpose computing on graphics processing units (GPGPU) (Owens et al., 2007) has gained prominence in a wide range of scientific fields, including medical imaging (Shams et al., 2010; Pratx and Xing, 2011; Eklund et al., 2013a) and neuroscience (Jeong et al., 2010; Pezoa et al., 2012; Ben-Shalom et al., 2013; Hoang et al., 2013; Yamazaki and Igarashi, 2013). The main reasons are that GPUs are inexpensive, power efficient and able to run several thousand threads in parallel, commonly providing a performance boost of 1–2 orders of magnitude for a small investment (see **Table 1**). Nonetheless, GPGPU is still uncommon in the neuroimaging field, where medical imaging and neuroscience intersect. Here, we therefore present BROCCOLI, a free software for parallel analysis of fMRI data on many-core CPUs and GPUs. BROCCOLI contains a large number of additions and improvements over our previous work (Eklund et al., 2010, 2011a; Forsberg et al., 2011; Eklund et al., 2012b). Some examples are Bayesian fMRI analysis using MCMC, first level statistical analysis using the Cochrane-Orcutt procedure (Cochrane and Orcutt, 1949), linear and non-linear registration for an arbitrary number of scales and support for *F*-tests as well as a larger number of regressors. While our previous implementations used CUDA, the most popular programming language for GPGPU, BROCCOLI is instead written in the open computing language (OpenCL) [see e.g., Munshi et al. (2011)]. This makes it possible to run BROCCOLI on many types of hardware, including CPUs, Nvidia GPUs, AMD GPUs, field programmable gate arrays (FPGAs), digital signal processors (DSPs) and other accelerators (e.g., the Intel Xeon Phi). As neuroimaging researchers use a wide range of operating systems (Hanke and Halchenko, 2011), it is also important that BROCCOLI can run efficiently regardless of the platform. One way to achieve this is to develop BROCCOLI for a specific platform (e.g., Windows), and then simply run BROCCOLI through a virtual machine for other platforms (e.g., Linux). However, direct access to GPU hardware through a virtual machine can currently be problematic, and was therefore not an option for our software. Instead, we have developed BROCCOLI using a combination of the platform-independent languages OpenCL and C++, and have made the source code freely available so that it can be compiled on any desired operating system supporting these widely deployed standards. In addition, as an added convenience, we have provided pre-compiled libraries for the Linux and Windows operating systems that can be linked to projects developed on either platform. A wrapper for Matlab is currently available, a Python wrapper is being developed and future plans include wrappers for bash and R. In addition to the improvements described above, BROCCOLI has also been extensively tested and compared to SPM, FSL, and AFNI by using a large number of freely available fMRI datasets. BROCCOLI is available as free software under GNU GPL3 and can be downloaded from github[4].

2. METHODS AND IMPLEMENTATION

The typical analysis pipeline for fMRI data is compromised of image registration, image segmentation, slice timing correction, smoothing, and statistical analyses. The methods used for these different processing steps in BROCCOLI are described in this section, and implementation details are given at the end of the section.

2.1. IMAGE REGISTRATION

Image registration for fMRI is used to align an anatomical T1 volume to a brain template (e.g., MNI or Talairach), to align an fMRI volume to the anatomical T1 volume, and to perform motion correction. The registration between the anatomical space and a standard brain space, often called spatial normalization, can be performed using a linear transformation model (e.g., affine or rigid) or by using a non-linear approach, which is much more computationally demanding. In a comparison of non-linear deformation algorithms for human brain MRI registration (Klein et al., 2009), the DARTEL algorithm in SPM took an average of 71 min to register a single T1 volume to the MNI template (1 mm^3 resolution) and the FNIRT algorithm in FSL used an average of 29 min. The AFNI software did not until recently have support for non-linear registration, but can now be achieved through the function 3dQwarp. Based on our benchmarking, non-linear registration with 3dQwarp

[4] https://github.com/wanderine/BROCCOLI/

Table 1 | Hardware configuration and performance measures of the computer used for testing the different software packages.

Device	Processor cores	Memory (GB)	Single precision (GFLOPS)	Double precision (GFLOPS)	Memory bandwidth (GB/s)	Price (USD)
Intel Core i7-3770K	4 (8 with hyper threading)	16	1 core: 56, 4 cores: 224	1 core: 28, 4 cores: 112	26	330
Nvidia GTX 680	1536	4	3090	129	192	500
AMD Radeon 7970	2048	3	3790	947	264	500

A Linux operating system was used (CentOS 6.4 64 bit) with an OCZ 128 GB SSD hard drive. The theoretical performance for single (32 bit floats) and double (64 bit floats) precision is given as giga floating point operations per second (GFLOPS). Prices are from newegg.com and should be seen as approximate.

takes about 36 min with a single-threaded version of AFNI, and 13 min using the multi-threaded OpenMP version (for a CPU running 8 threads). Thus, depending on the algorithm, normalization for a study involving 30 subjects can take 5–35.5 h. Moreover, to obtain satisfactory results, it may be necessary to run the registration algorithm with a number of different settings. For these reasons, affine registration to a standard brain space is sometimes performed instead of a non-linear one, even though the non-linear approach can yield a better registration. Another time saving approach is to perform spatial normalization to a brain template of lower resolution, e.g., 2 mm^3 voxels, but this solution is less appealing, since spatial resolution is sacrificed. Due to the computational challenges of image registration, GPU acceleration of such algorithms is very popular with some 60 publications since 1998 (Shams et al., 2010; Fluck et al., 2011; Pratx and Xing, 2011; Eklund et al., 2013a). GPUs can thus easily be used in the neuroimaging field, to for example enable more widespread use of demanding non-linear registration algorithms.

2.1.1. Linear image registration

BROCCOLI uses a single registration algorithm to perform the three described registrations (T1-to-MNI,fMRI-to-T1, and motion correction). Here we summarize the algorithm, which has been previously described (Eklund et al., 2010). The main idea of the algorithm is to use the optical flow equation (Horn and Schunck, 1981)

$$\nabla I^T \boldsymbol{v} = \Delta I, \tag{1}$$

where ∇I is the gradient of the volume, $\boldsymbol{v}$ is a motion vector that describes the difference between the volumes and ΔI is the intensity difference between the two volumes. The aperture problem, however, prevents us from solving this equation directly, as there are three unknown variables (the motion in x, y, and z), but only one equation. Instead of solving the equation for each voxel separately, one can minimize the expression over the entire volume. The total squared error can be written as

$$\epsilon^2 = \sum_i \left(\nabla I(\boldsymbol{x}_i)^T \boldsymbol{v}(\boldsymbol{x}_i) - \Delta I(\boldsymbol{x}_i) \right)^2, \tag{2}$$

where $\boldsymbol{x}_i$ denotes the position of voxel i. A linear model of the motion field can be used to represent a motion vector in each voxel. The motion field $\boldsymbol{v}(\boldsymbol{x})$ for affine transformations in 3D can be modelled with a 12-dimensional parameter vector, $\boldsymbol{p} = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}]^T$, and a base matrix $\boldsymbol{B}(\boldsymbol{x})$ according to (Hemmendorff et al., 2002)

$$\boldsymbol{v}(\boldsymbol{x}) = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + \begin{bmatrix} p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \\ p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 & x & y & z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x & y & z & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & x & y & z \end{bmatrix}}_{\boldsymbol{B}} \boldsymbol{p}.$$

The first three parameters are the translations and the last nine parameters form a transformation matrix (if an identity matrix is added, as the parameter vector $\boldsymbol{p}$ used here only describes the difference between the two volumes). The variables x, y, and z are the coordinates of voxel $\boldsymbol{x}$. By using the model of the motion field, $\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\, \boldsymbol{p}$, the error measure can be written as

$$\epsilon^2 = \sum_i \left(\nabla I(\boldsymbol{x}_i)^T \boldsymbol{B}(\boldsymbol{x}_i)\, \boldsymbol{p} - \Delta I(\boldsymbol{x}_i) \right)^2. \tag{4}$$

The derivative of this expression, with respect to the parameter vector, is given by

$$\frac{\partial \epsilon^2}{\partial \boldsymbol{p}} = 2 \sum_i \boldsymbol{B}_i^T \nabla I_i \left(\nabla I_i^T \boldsymbol{B}_i\, \boldsymbol{p} - \Delta I_i \right), \tag{5}$$

and setting the derivative to zero yields the following linear equation system

$$\underbrace{\sum_i \boldsymbol{B}_i^T \nabla I_i \nabla I_i^T \boldsymbol{B}_i}_{\boldsymbol{A}} \boldsymbol{p} = \underbrace{\sum_i \boldsymbol{B}_i^T \nabla I_i \Delta I_i}_{\boldsymbol{h}}, \tag{6}$$

where $\boldsymbol{A}$ is a matrix of size 12 × 12 and $\boldsymbol{h}$ is a vector of size 12 × 1. The best parameter vector can finally be calculated as

$$\boldsymbol{p} = \boldsymbol{A}^{-1} \boldsymbol{h}. \tag{7}$$

The system of linear equations is easy to solve, while the computationally demanding part is to sum over all voxels. L_2 norm minimization makes it possible to calculate the parameters that give the best solution. The solution can then be improved by iterating the algorithm and accumulating the parameter vector (to avoid repeated interpolation). The most common approach is otherwise to maximize a similarity measure by searching for the best parameters, using some optimization algorithm. To handle large differences between two volumes, it is common to start the registration on a coarse scale and then improve the registration by moving to finer scales. BROCCOLI uses three to four scales for the registration between T1 and MNI and between fMRI and T1; the difference between each scale is a factor two in each dimension.

The estimated affine transformation parameters can be restricted to a rigid transformation (i.e., translations and rotations only), and is accomplished in BROCCOLI by applying a singular value decomposition (SVD) to the transformation matrix and then forcing the singular values to be one. Rigid registration is used for fMRI-T1 registration and for motion correction, while affine registration (12 parameters) is used for the T1-MNI registration. For the motion correction procedure, the rotation angles $\theta_1, \theta_2, \theta_3$ are extracted from the estimated rotation matrix for each time point using the following formulas (Shoemake, 1994; Day, 2012)

$$\begin{aligned}\theta_1 &= atan2(p_9, p_{12}),\\ c_2 &= \sqrt{p_4 \cdot p_4 + p_5 \cdot p_5},\\ \theta_2 &= atan2(-p_6, c_2),\\ s_1 &= sin(\theta_1),\\ c_1 &= cos(\theta_1),\\ \theta_3 &= atan2(s_1 \cdot p_{10} - c_1 \cdot p_7, c_1 \cdot p_8 - s_1 \cdot p_{11}),\end{aligned} \tag{8}$$

where $atan2(a, \ b)$ is the four quadrant arctangent of a and b. The main reasons for extracting the rotation angles are to use them as nuisance regressors in the statistical analysis and to present them to the user.

2.1.2. Non-intensity based image registration

The registration algorithm used in BROCCOLI is not based on the image intensity directly, e.g., the image gradient as described above. Instead, the algorithm is based on matching edges to edges and lines to lines, by using the concept of local phase from quadrature filter responses (Granlund and Knutsson, 1995; Knutsson and Andersson, 2003). A quadrature filter is complex valued in the spatial domain; the real part is a line detector and the imaginary part is an edge detector. The local phase is the relationship between the real and imaginary filter responses and describes the type of local structure (e.g., a line or an edge), while the magnitude can be seen as a certainty measure of how likely it is that the filter detected a structure. The local phase concept is illustrated in **Figure 1**. The quadrature filters need to be created using filter optimization techniques, which simultaneously consider properties in the spatial domain and the frequency domain (Granlund and Knutsson, 1995; Knutsson et al., 1999). In the presented equations, the image gradient ∇I is replaced with a phase gradient $\nabla\varphi$ and the image difference ΔI is replaced with a phase difference $\Delta\varphi$. The phase difference can be calculated as

$$\Delta\varphi = arg\left(q_1 \cdot q_2^*\right), \tag{9}$$

where q_1 and q_2 are the complex valued quadrature filter responses for the two volumes and $*$ denotes complex conjugation. A nice property of the local phase is that it is invariant to the image intensity (all edges are for example interpreted equally, regardless if the image intensity changes from 0 to 1 or from 10 to 11), making it easier to register volumes from different modalities or volumes with different or varying contrast. Phase based optical flow was introduced in the field of computer vision (Fleet and Jepson, 1990) and eventually propagated to the medical imaging domain (Hemmendorff et al., 2002; Knutsson and Andersson, 2005; Mellor and Brady, 2005). While phase based image registration can in some cases be more robust against intensity differences (Hemmendorff et al., 2002; Mellor and Brady, 2005; Eklund et al., 2011b), a drawback is that it requires filtering with a number of (non-separable) filters in each iteration, which is computationally demanding. Fortunately, GPUs are perfectly suited for parallel operations like filtering (Eklund and Dufort, 2014).

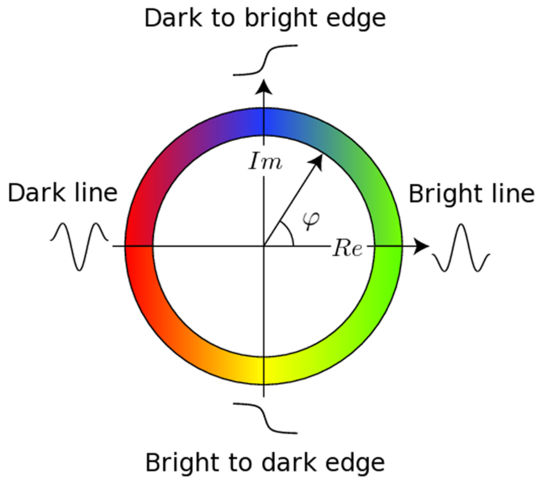


FIGURE 1 | This figure presents the main concept of local phase φ from quadrature filter responses. A quadrature filter is complex valued in the spatial domain; the real part is a line detector and the imaginary part is an edge detector. If the filter response only contains a real valued component, it means that the filter detected a line. If the filter response only contains an imaginary valued component, it means that the filter detected an edge. It is important to combine the local phase with the magnitude of the complex valued filter response, as the local phase does not have any meaning for a low magnitude.

2.1.3. Non-linear image registration

As previously mentioned, non-linear methods can lead to a significantly better registration between a subject specific anatomical volume and a brain template. BROCCOLI uses the Morphon (Knutsson and Andersson, 2005; Forsberg et al., 2011; Forsberg, 2013) to perform non-linear registration. The Morphon is also based on phase based optical flow, and the two most important parts of the Morphon are, therefore, the same as for the linear registration algorithm; to apply a number of quadrature filters and to calculate phase differences. The main differences are that the linear algorithm uses three quadrature filters (oriented along x, y, and z) and solves one equation system for the entire volume, while the Morphon uses six quadrature filters (evenly distributed on the half sphere of an icosahedron) and solves as many equation systems as there are voxels. The error being minimized in each voxel can be written as

$$\epsilon^2 = \sum_{k=1}^{N}\left(c_k \boldsymbol{T}\left(\Delta\varphi_k \hat{\boldsymbol{n}}_k - \boldsymbol{d}\right)\right)^2, \tag{10}$$

where $\Delta\varphi_k$ is the phase difference between the two volumes for quadrature filter k, c_k is a certainty estimate for filter k, $\hat{\boldsymbol{n}}_k$ is the orientation vector for filter k, N is the number of quadrature filters, $\boldsymbol{d}$ is the displacement vector to be optimized and $\boldsymbol{T}$ is a local structure tensor (Knutsson, 1989; Granlund and Knutsson, 1995; Knutsson et al., 2011). A local structure tensor in image processing is analogous to a diffusion tensor in diffusion tensor imaging (DTI); it represents the magnitude and orientation of the signal in each neighborhood. The tensor can be calculated from the six complex valued quadrature filter responses as (Granlund and Knutsson, 1995)

$$\boldsymbol{T} = \sum_{k=1}^{N} |q_k|\left(\frac{5}{4}\hat{\boldsymbol{n}}_k\hat{\boldsymbol{n}}_k^T - \frac{1}{4}\boldsymbol{I}\right), \tag{11}$$

where $\boldsymbol{I}$ is an identity tensor. The purpose of using the tensor in the error measure is to reinforce displacement estimates along the local predominant orientations (i.e., displacements perpendicular to edges and lines). Using an L_2-norm, the best displacement vector can be calculated for each voxel directly, by once again solving a linear system (of size 3 x 3), i.e.,

$$\boldsymbol{d} = \left(\sum_{k=1}^{N} c_k^2 \boldsymbol{T}^T \boldsymbol{T} \right)^{-1} \sum_{k=1}^{N} c_k^2 \Delta\varphi_k \boldsymbol{T}^T \boldsymbol{T} \hat{\boldsymbol{n}}_k. \tag{12}$$

The estimated displacement field is regularized by applying Gaussian smoothing separately to each motion component (x, y, z) before it is used to warp the T1 volume. Just as for the linear registration, the displacement field is accumulated in each iteration to avoid repeated interpolation. An affine registration (12 parameters) is first estimated between the T1 volume and the MNI template before estimation of the non-linear displacement field.

2.2. IMAGE SEGMENTATION

SPM has several functions for segmenting brain volumes. FSL provides BET (brain extraction tool) and FAST (FMRIB's automated segmentation tool) while AFNI provides the function 3dSkullStrip. BROCCOLI performs skullstripping by first registering the T1 volume to MNI space, using an MNI template with skull, then applies an inverse transform to the MNI brain mask and finally performs a multiplication between the transformed mask and the original T1 volume to obtain a skullstripped version of the T1 volume. The skullstripped T1 volume is then aligned to an MNI template without skull, to improve the alignment, and the MNI brain mask is again inversely transformed (using the new registration parameters) and multiplied with the original T1 volume, to obtain a better skullstrip. The fMRI data is segmented by first applying 4 mm 3D Gaussian smoothing to one of the fMRI volumes and then using a threshold that is 90% of the mean value.

2.3. SLICE TIMING CORRECTION

Slice timing correction is normally applied to fMRI data (Sladky et al., 2011), as the slices in each volume are collected at slightly different time points. BROCCOLI sets the middle slice as the reference and then applies cubic interpolation in time to correct for the temporal difference between the slices.

2.4. SMOOTHING

fMRI data is frequently spatially smoothed. The non-linear registration algorithm also uses Gaussian smoothing, for example to reguralize the tensor components and the resulting displacement field in each iteration. BROCCOLI utilizes a simple form of normalized convolution (Knutsson and Westin, 1993), called normalized averaging, to avoid problems with voxels close to the edge of the brain being influenced by voxels outside the brain. The normalized filter response *nfr* is calculated as

$$nfr = \frac{(v \cdot c) * f}{c * f}, \tag{13}$$

where f is the filter, v is one fMRI volume, c is a certainty measure, $*$ denotes convolution and $\cdot$ denotes pointwise multiplication. The certainty is simply the fMRI brain mask, such that the certainty is one inside the brain and zero outside. If a gray matter segmentation is available, the same approach can be used to prevent similar problems with smoothing that includes values from other types of brain matter (by setting the certainty to one for gray voxels and zero for all other voxels).

2.5. STATISTICAL ANALYSIS

The statistical analysis is the core of all fMRI software packages. The use of GPUs for statistical computations is a relatively new concept (Suchard et al., 2010; Guo, 2012) and can for example be used to speedup demanding Markov Chain Monte Carlo (MCMC) simulations (Lee et al., 2010). We believe that GPUs (or at least the computational capacity they confer) are a necessary component for incorporation of developments in the field of statistics to the field of neuroimaging, especially for high resolution fMRI data (Feinberg and Yacoub, 2012). By using GPUs, computationally demanding non-parametric tests can be used instead of parametric ones (Nichols and Holmes, 2002; Eklund et al., 2011a) and MCMC based methods [e.g., (Woolrich et al., 2004)] also become feasible (da Silva, 2011).

The SPM, FSL, and AFNI software packages are all mainly based on the general linear model (GLM) for first (subject) and second level (group) analyses, as proposed by Friston et al. (1994). The GLM can be written in matrix form as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{14}$$

where $\boldsymbol{y}$ are the observations for one voxel, $\boldsymbol{\beta}$ are the parameters to estimate, $\boldsymbol{X}$ is the design matrix (model) containing all the regressors and $\boldsymbol{\epsilon}$ are the errors that cannot be explained by the model. As the GLM is applied to each voxel independently, it is perfectly suited for parallel implementations. By minimizing the squared error $||\boldsymbol{\epsilon}||^2$, it can be shown that the best parameters (for independent errors) are given by

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T \boldsymbol{X} \right)^{-1} \boldsymbol{X}^T \boldsymbol{y}. \tag{15}$$

A useful property of this expression is that the term $\left(\boldsymbol{X}^T \boldsymbol{X} \right)^{-1} \boldsymbol{X}^T$ is the same for all voxels and can, thus, be precalculated. A t-test value can easily be calculated from the estimated weights as

$$t = \frac{\boldsymbol{c}^T \hat{\boldsymbol{\beta}} - u}{\sqrt{\mathrm{var}\left(\hat{\boldsymbol{\epsilon}}\right) \ \boldsymbol{c}^{\mathrm{T}} \left(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X} \right)^{-1} \boldsymbol{c}}}, \tag{16}$$

where $\boldsymbol{c}$ is a contrast vector, $\hat{\boldsymbol{\epsilon}}$ is the residual of the GLM and u is a scalar for the null hypothesis $\boldsymbol{c}^T \hat{\boldsymbol{\beta}} = u$. An F-test value can in a similar manner be calculated as

$$F = \frac{\left(\boldsymbol{C}\hat{\boldsymbol{\beta}} - \boldsymbol{u} \right)^T \left(\mathrm{var}\left(\hat{\boldsymbol{\epsilon}}\right) \ \boldsymbol{C} \left(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X} \right)^{-1} \boldsymbol{C}^{\mathrm{T}} \right)^{-1} \left(\boldsymbol{C}\hat{\boldsymbol{\beta}} - \boldsymbol{u} \right)}{N}, \tag{17}$$

where $\boldsymbol{C}$ is a contrast matrix and N is the number of contrasts.

2.5.1. First level analysis

The first level fMRI analysis starts with slice timing correction and motion correction. The estimated motion parameters (translations and rotations) are included in BROCCOLI by default as additional regressors in the GLM design matrix, to further reduce effects of head motion (Johnstone et al., 2006). Gaussian smoothing is applied to each fMRI volume and the GLM is finally applied to the smoothed volumes. In addition to motion regressors and regressors for the experimental design, the design matrix in BROCCOLI also contains regressors to remove the mean and trends that are linear, quadratic or cubic. The effect of using these additional regressors is similar to a highpass filtering. The GLM errors are for first level fMRI analysis often modelled as an auto regressive (AR) process,

$$\epsilon_t = \sum_{i=1}^{p} \rho_i \epsilon_{t-i} + w_t, \tag{18}$$

where p is the order of the AR process, ρ_i are the AR parameters and w is white noise with variance σ^2. A Cochrane-Orcutt procedure (Cochrane and Orcutt, 1949) is used in BROCCOLI to estimate the beta weights for autocorrelated errors. The GLM weights $\boldsymbol{\beta}$ are first estimated using ordinary least squares (equation 15) and then a voxel-wise AR model of the fourth order is used to model the residuals (Worsley et al., 2002). The AR parameters are estimated by solving the Yule-Walker equations independently for each voxel. Each volume of AR estimates is spatially smoothed with a 7 mm Gaussian filter to further improve the estimates (Woolrich et al., 2001; Worsley et al., 2002; Gautama and Hulle, 2004), before the actual whitening is applied to the smoothed fMRI data and the regressors in the design matrix (such that each voxel gets its own specific design matrix). The components of the whitened data $\tilde{\boldsymbol{y}}$ and the whitened regressors $\tilde{\boldsymbol{X}}$ are thus calculated as

$$\tilde{y}_t = y_t - \sum_{i=1}^{4} \rho_i y_{t-i}, \tag{19}$$

$$\tilde{X}_{t,\, r} = X_{t,\, r} - \sum_{i=1}^{4} \rho_i X_{t-i,\, r}, \tag{20}$$

where ρ_i are the spatially smoothed AR estimates, r denotes regressor and t denotes time point. The whitened data $\tilde{\boldsymbol{y}}$ and the whitened regressors $\tilde{\boldsymbol{X}}$ are then used to estimate new beta weights, according to

$$\tilde{\boldsymbol{\beta}} = \left(\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}}\right)^{-1} \tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{y}}. \tag{21}$$

As a last step, the AR parameters are re-estimated using residuals calculated with the new weights $\tilde{\boldsymbol{\beta}}$, the original data $\boldsymbol{y}$ and the original regressors $\boldsymbol{X}$. The Cochrane-Orcutt procedure is repeated three times to obtain good estimates of the GLM weights and the AR parameters. Finally, the statistical maps are calculated using the variance of the uncorrelated residuals $\tilde{\boldsymbol{\epsilon}}$, obtained as

$$\tilde{\boldsymbol{\epsilon}} = \tilde{\boldsymbol{y}} - \tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}. \tag{22}$$

FSL uses a similar iterative approach to estimate a voxel-wise prewhitening matrix (Woolrich et al., 2001), with the exception that the spatial smoothing is done separately for different tissue types. The voxel-specific noise model used in BROCCOLI has been shown to yield more valid results than those obtained from SPM (Eklund et al., 2012a), which uses a global AR(1) model. After the first level statistical analysis, the results (e.g., beta weights) are transformed to MNI space, by combining the estimated registration parameters for T1-to-MNI and fMRI-to-T1 transformations and the estimated displacement field from the non-linear registration.

2.5.2. Second level analysis

The second level analysis in fMRI is straightforward compared to the first level analysis, once all the first level results are in a common brain space. A group-wise t-test or F-test can easily be performed by using the same functions as for the first level GLM. BROCCOLI currently only supports conventional t-tests and F-tests for second level analysis, but we plan to also include other types of analyses (e.g., where the variance of the beta estimates are used as weights) in future releases.

2.5.3. Frequentist inference

In contrast to other software packages for fMRI analysis, BROCCOLI is not based on parametric statistics. All p-values are instead calculated through non-parametric permutation tests (Dwass, 1957; Nichols and Holmes, 2002), both for first level and second level analyses. The main motivation is that parametric statistics require several assumptions to be met for the results to be valid. In fMRI it is also necessary to correct for a large number of tests, due to the high spatial resolution. The multiple testing makes the parametric assumptions much more critical, as one has to move far along the tail of the null distribution. The SPM software relies on Gaussian random field theory (GRFT) to correct for the multiple testing (Worsley et al., 1992), while FSL mainly works with GRFT and non-parametric permutation tests (for group analyses only). AFNI instead uses the false discovery rate (FDR) (Genovese et al., 2002) and a cluster simulation tool. A permutation test solves the problem of multiple testing in a very simple way. In each permutation, only the largest value of the statistical map (e.g., the maximum t-test value, the maximum F-test value, the size or mass of the largest cluster etc.) is saved to form the null distribution of the maximum test statistics. Corrected p-values are finally calculated as the proportion of values in the estimated null distribution that are larger than or equal to the test value for the current voxel or cluster. A threshold for a certain significance level α, corrected for multiple testing, can be calculated by first sorting the estimated null distribution values, and then simply using the value that is larger than $(100 - \alpha)$ % of the values. The main problem is that a large number of permutations, normally 1000–10,000, are required to obtain a good estimate of the null distribution. Since a full statistical analysis needs to be performed in each permutation, the total processing time can be several hours or days for a single test, using conventional multi-core CPU implementations. This is the main reason why permutation tests are not standard procedure in the neuroimaging field.

For first level analysis in BROCCOLI, detrending and whitening [using a voxel-wise AR(4) model as previously described] is applied to the motion corrected data and new fMRI data is then generated in each permutation by using an inverse whitening transform with randomly permuted whitened data. The smoothing has to be applied in each permutation, as the smoothing alters the autocorrelation structure of the fMRI data. Permutation testing for the second level analysis is much easier, as no whitening or smoothing is required. See our previous work for further information on the non-parametric analysis (Eklund et al., 2011a, 2012a).

2.5.4. *Bayesian inference*

The GLM model previously described can alternatively be analyzed using Bayesian methods. A Bayesian analysis begins with a prior distribution $p(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\rho})$ over the model parameters and subsequently updates the prior with the observed data. The result is the posterior distribution $p(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\rho}|\boldsymbol{X}, \boldsymbol{y})$, which encapsulates all information about the unknown parameters conditional on the observed data. In fMRI, the brain activity can be visualized as a heat map of $\Pr(\beta_i > 0|\boldsymbol{X}, \boldsymbol{y})$, commonly known as a posterior probability map (PPM) (Friston et al., 2002). The joint posterior $p(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\rho}|\boldsymbol{X}, \boldsymbol{y})$ is often not tractable in analytical form, but can be approximated by different approaches. The most common approach in the fMRI field is to use approximation techniques like variational Bayes, where the posterior is factorized into several independent factors to obtain an analytical expression (Penny et al., 2003). A less common approach is to use techniques based on Markov Chain Monte Carlo (MCMC) simulation. MCMC produces a sample from the posterior, and the probability of activity $\Pr(\beta_i > 0|\boldsymbol{X}, \boldsymbol{y})$ can be approximated by the proportion of simulated β_i being larger than zero. The PPM for any contrast is also directly available from the posterior simulations. Note that since simulations are done using the joint posterior, PPMs are not conditional on point estimates of σ^2 and $\boldsymbol{\rho}$, leading to more accurate inferences regarding brain acitivity.

BROCCOLI uses a specific MCMC algorithm, the Gibbs sampler, to generate draws from the posterior by iteratively simulating from two full conditional posteriors. First, the autocorrelation parameters $\boldsymbol{\rho}$ are updated by simulation from $\boldsymbol{\rho}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{y}, \boldsymbol{X}$ as a (multivariate) Gaussian distribution. Second, the variance σ^2 is updated by simulation from $\sigma^2|\boldsymbol{\rho}, \boldsymbol{y}, \boldsymbol{X}$ as an inverse Gamma distribution and the GLM weights $\boldsymbol{\beta}$ are finally updated by simulation from $\boldsymbol{\beta}|\sigma^2, \boldsymbol{\rho}, \boldsymbol{y}, \boldsymbol{X}$ as a (multivariate) Gaussian distribution. These conditional distributions are obtained when the priors for $\boldsymbol{\beta}|\sigma^2$ and $\boldsymbol{\rho}$ are Gaussian and the prior on σ^2 is inverse Gamma. The exact details of each updating step can be found in most Bayesian textbooks, see e.g., Murphy (2012). Note that each updating step conditions on the most recently simulated value for the conditioning parameters. While MCMC methods can theoretically be used to approximate any posterior, a common problem is the significantly longer processing time compared to techniques like variational Bayes. BROCCOLI runs a large number of MCMC chains in parallel to reduce the processing time.

2.6. IMPLEMENTATION

We will here describe the implementation of BROCCOLI for the different algorithms. Readers are referred elsewhere for introductions to GPU programming (Kirk and Hwu, 2010; Munshi et al., 2011; Sanders and Kandrot, 2011). Most of the OpenCL code uses single precision to achieve maximum performance, while some host code uses double precision (to for example obtain the optimal affine registration parameter vector). The open source library Eigen[5, 6] is used in BROCCOLI to perform matrix calculations on the host.[7]

2.6.1. *Image registration*

The described linear and non-linear registration algorithms are easy to run in parallel. The filtering operation applied in each iteration is the most demanding part, especially since quadrature filters are non-separable, and has therefore been carefully optimized. Filtering can be performed as a multiplication in the frequency domain, after the application of a fast Fourier transform (FFT) to the signal and the filter, or as a convolution in the spatial domain. BROCCOLI uses the convolution approach, for three reasons. First, the FFT approach requires an FFT library while the convolution approach can rather easily be implemented manually. The CUDA programming language provides the CUFFT library, and a similar OpenCL library called clFFT has recently appeared. However, clFFT is in our opinion not yet as mature as CUFFT. The user, for example, has to compile the whole project to obtain a library file. Second, a convolution approach often provides high performance over a wide range of data sizes, while an FFT normally performs best for data sizes being a power of 2. Third, the convolution approach is less memory demanding as the FFT approach requires that the filters are stored as the size of the signal for an elementwise multiplication.

Convolution is easy to run in parallel, and high performance can be achieved by taking advantage of the fact that the filter responses for neighboring voxels use mainly the same input data. An easy way to implement a non-separable 3D convolution is to take advantage of the texture memory, as the texture memory cache can be used to speedup reads that are spatially local. Such an implementation will, however, be limited by the global memory bandwidth. A better approach is to take advantage of the local memory[8] available in modern GPUs (CPUs do not normally have local memory physically; it can instead be simulated by the OpenCL driver). By first reading values from global memory into local memory, all the threads in a thread block can repeatedly read from the local memory very efficiently. The Nvidia GTX 680 has 48 KB of local memory per multiprocessor; it can for example store a 3D array of 32 × 32 × 12 float values. The quadrature filters used in BROCCOLI contain 7 × 7 × 7 coefficients, only 26 × 26 × 6 = 4,056 filter responses will therefore be valid for each multiprocessor. The reason for this is that the convolution is undefined along a boundary of $(N - 1)/2$ pixels for an N ×

[5] http://eigen.tuxfamily.org/index.php?title=Main_Page

[6] https://bitbucket.org/eigen/eigen/

[7] For readers not familiar with GPU programming, the CPU is often called the host while the GPU is called the device.

[8] Local memory in OpenCL is the same thing as shared memory in CUDA.

N kernel. The yellow pixels in **Figure 2** illustrate the invalid filter responses along the image borders for a filter size of 7 × 7. To maximize the number of valid filter responses per multiprocessor, a better approach to non-separable 3D convolution is to instead perform non-separable 2D convolution on the GPU, and then accumulate the filter responses by calling the convolution kernel for each slice of the filter [i.e., instead of running all 6 for-loops (three for the data and three for the filter) on the GPU, run 5 on the GPU and 1 on the CPU]. The local memory can for 2D be used to store two arrays of 96 × 64 float values, which instead gives a total of 10,440 valid filter responses per multiprocessor (two blocks of 90 × 58 pixels). The reason for using two arrays instead of one, is that each multiprocessor on the Nvidia GTX 680 can concurrently run 2048 threads, but only 1024 threads per thread block. The 1024 threads per block are arranged as 32 along the x-direction and 32 along the y-direction, to for example fit the number of local memory banks (32). Each thread starts by reading 6 values from global memory into local memory (96 × 64 / 1024 = 6) and then calculates 2 filter responses (giving two 32 × 32 blocks). Three additional filter responses are then calculated by most of the threads, yielding two blocks of 32 × 26 pixels and one block of 26 × 32 pixels. Finally, a number of threads are used to calculate the final 26 × 26 filter responses. The usage of local memory for non-separable 2D convolution is illustrated in **Figure 2**. As several quadrature filters need to be applied to the two volumes being registered (3 for linear registration and 6 for non-linear registration), 3 filters are applied simultaneosly once the data has been loaded into local memory. To achieve maximum performance, the for-loops have been unrolled manually using a Matlab script. To run a short for-loop on a GPU can result in a sub-optimal performance, as it can take a longer time to setup the for-loop than to run it (this is especially true for nested for-loops). The filters are stored in constant memory, as they are used by all threads and since each multiprocessor has a constant memory cache. The filter responses are stored in thread specific registers. Note that calculating 6 filter responses per thread results in a much better ratio of memory operations and calculations, compared to a straight forward approach using texture memory (where each thread calculates a single filter response). Interested readers are referred to our previous work (Eklund and Dufort, 2014) and our separate github repository[9] for further details. The AMD GPU and the Intel CPU used in our case have only 32 KB of local memory, the AMD GPU can also only run 256 threads per thread block. The code for these devices instead uses one local memory array of 128 × 64 pixels and calculates 120 × 58 filter responses in blocks of 16 × 16 pixels.

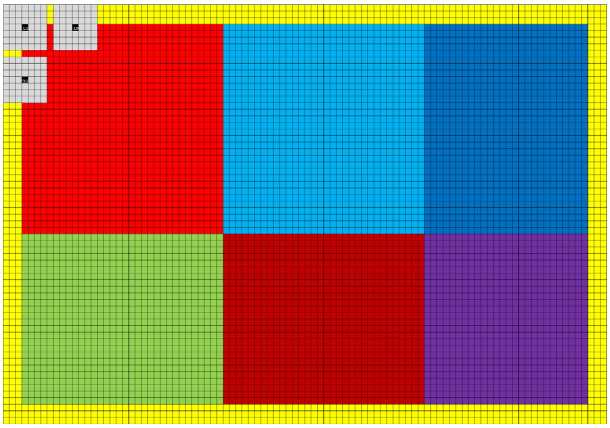


FIGURE 2 | The grid represents 96 × 64 pixels in local memory (each square is one pixel). As 32 x 32 threads are used per thread block, each thread needs to read 6 values from global memory into local memory [(96 × 64)/(32 × 32) = 6]. A yellow halo needs to be loaded into local memory to be able to calculate all the filter responses. In this case 90 × 58 valid filter responses are calculated, making it possible to apply at most a filter of size 7 × 7. The 90 × 58 filter responses are calculated as 6 runs, the first 2 consisting of 32 × 32 pixels (marked light red and light blue). The 1024 filter responses (32 × 32) are calculated in parallel, and the gray squares represent three filter responses being calculated. Note that neighboring filter responses are calculated using mainly the same pixels. Three additional filter responses are calculated in blocks of 32 × 26 or 26 × 32 pixels (marked green, dark blue and dark red). Finally, a block of 26 × 26 pixels is processed (marked purple). The halo can easily be changed to handle larger filters.

The linear registration algorithm involves a summation over all voxels to setup an equation system (equation 6). BROCCOLI performs this summation using three kernels. The first kernel performs all the necessary multiplications and each thread calculates the sum for one voxel along the x-direction. The number of threads per thread block is equal to the width of the volume. The second kernel continues the summation along the y-direction (the number of threads per block is set to the depth of the volume) and the third kernel sums along z. The resulting equation system is finally copied to the host, to calculate the best parameter vector.

Except for the filtering and the summation operation, the other required functions are straight forward to implement. For the linear registration algorithm, one kernel is used in BROCCOLI to calculate phase differences (equation 9) and certainties and three kernels are used to calculate phase gradients $\nabla\varphi$ along x, y and z (Eklund et al., 2010). For the non-linear registration algorithm, one kernel is used to calculate the tensor components (equation 11), one kernel is used to setup the equation system in each voxel and one kernel solves the equation system (equation 12). Both the linear and the non-linear registration algorithm use one additional kernel to interpolate from the volume being moved to match the template. The texture memory is used for these two kernels, as it has hardware support for linear interpolation in 1, 2, and 3 dimensions. For all these kernels, each thread performs the operations for one voxel. To make sure that the same code runs on both Nvidia and AMD GPUs, 256 threads per block are used.

2.6.2. Smoothing

The smoothing operation is also implemented as a convolution. As the Gaussian smoothing filters are Cartesian separable, three kernels are used to smooth along x, y, and z. Similarly to the non-separable convolution, local memory is used to obtain a more efficient implementation. The details of how the separable smoothing is performed will therefore not be given here.

[9] https://github.com/wanderine/NonSeparableFilteringCUDA

2.6.3. Statistical analysis

The statistical analysis of fMRI data is perfect for parallel processing; each thread performs the required calculations for one voxel. Just as for the registration kernels, all the statistical kernels use 256 threads per block to fit both Nvidia and AMD GPUs. For first level analysis assuming independent errors and for second level analysis, the pseudo inverse of the design matrix [i.e., $\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$] is calculated on the host and stored in constant memory (as it is the same for all voxels). Calculation of the beta weights for one voxel can then be simply performed as a number of scalar products between the rows of the pseudo inverse and the data points of the current voxel (see equation 15). The resulting beta weights are stored as registers in each thread. However, each GPU thread can only handle a limited number of variables, BROCCOLI currently therefore supports a maximum of 25 regressors. To simply loop over the number of regressors may result in suboptimal performance, for two reasons. The first reason is that if the index to the beta array is not known at compile time, e.g., beta[i], the compiler may put beta in global memory instead of registers. The second reason is that short for-loops are inefficient on GPUs (as mentioned in the filtering implementation). For optimal performance, BROCCOLI instead uses a switch-case approach to first determine the number of regressors being used. The code for each case is also unrolled, such that all accesses to the beta array are known at compile time. To calculate the t-test or F-test value efficiently in each voxel, some additional values, e.g., $\boldsymbol{c}^T\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{c}$ from equation 16, are also pre-calculated and stored in constant memory. A limitation of the described approach is that the constant memory is normally only 32–128 KB; it can thus not store arbitrary large design matrices. A potential solution to this problem is to instead use texture memory, and take advantage of the texture memory cache instead of the constant memory cache.

The Cochrane-Orcutt procedure is harder to implement, as each voxel then uses a specific design matrix (after whitening according to equation 20). To calculate a pseudo inverse in each thread is problematic, as a design matrix for first level analysis easily can contain 200 timepoints and 15 regressors. Such an operation would thus require at least 3000 floats per thread, far outstripping the capabilities of some contemporary devices. For example, the Nvidia GTX 680 can handle only 63 floats per thread in its registers. Additional floats will spill into slow global memory (called local memory in CUDA), which may degrade the performance significantly. GPUs that have a L1 and/or L2 cache may be able to still use a larger number of registers efficiently. A possible solution could be to instead use the updating formula derived for MCMC (equation 24), but such an approach can also require a large number of registers [e.g., 40 registers for the $\boldsymbol{m}_{ij}$ variables for 10 regressors and an AR(1) model]. The current solution is to instead calculate all the pseudo inverses on the host and then copy them to slow global memory. For these reasons, the Cochrane-Orcutt procedure is not yet optimized in terms of speed. Permutation testing for first level analysis therefore currently uses the simpler approach assuming independent errors. The permutation based p-values will still be valid, as the same analysis is applied in each permutation (whitening is applied prior to the permutations, and the autocorrelation is then put back in each permutation).

The whitening operation that is applied prior to the single subject permutations, and in the Cochrane-Orcutt procedure, requires that an AR model is estimated for each voxel. To accomplish this, each thread loops over time and sets up the Yule-Walker system of equations for one voxel. The AR(4) parameters are then calculated by directly solving these equations using a matrix inverse. One limitation of this approach is that more advanced AR models [e.g., an AR(8) model] requires a larger number of registers, both to store the parameters and to calculate the matrix inverse. For the inverse whitening applied in each permutation, to generate new data, all the threads also loop over time to generate new time series.

Permutation tests involving cluster based inference require that a clustering operation is performed in each permutation, to calculate the extent or mass of the largest cluster. BROCCOLI uses the parallel label equivalence algorithm proposed by Hawick et al. (2010) for this purpose. The algorithm is implemented as five kernels. The first kernel assigns an unique starting label to each voxel that survives the initial voxel-wise threshold (e.g., $p = 0.01$, uncorrected for multiple comparisons). In the second kernel each voxel checks its 26 neighbors to see if there is a label with a lower value. If a lower label is found, the label of the center voxel is updated and an update flag is set to 1. The third kernel resolves label equivalences, in order to minimize the number of times the second kernel has to be launched [see Hawick et al. (2010) for details]. The second and third kernels are launched repeatedly, until the update flag is no longer set to 1. To calculate the size of each cluster, a fourth kernel is applied where each thread atomically increments a cluster specific counter (determined by the cluster label). Finally, a fifth kernel is used to obtain the size of the largest cluster; the implementation relies on the atomic max operation.

The Bayesian MCMC algorithm can with careful memory management lead to a substantial time reduction compared to a sequential approach. To see the importance of memory management, consider simulating from the full conditional posterior of $\boldsymbol{\beta}$ and σ^2. Conditional on $\boldsymbol{\rho}$, this is a standard linear regression update on the transformed model

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\epsilon}}, \tag{23}$$

where $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{y}}$ are obtained by pre-whitening $\boldsymbol{X}$ and $\boldsymbol{y}$ with the most recently simulated coefficients in $\boldsymbol{\rho}$ (as described in equations 19 and 20). Since $\boldsymbol{\rho}$ changes in every $\boldsymbol{\rho}$-update, both $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{y}}$ need to re-computed in each iteration of the Gibbs sampler. Both $\boldsymbol{X}$ and $\boldsymbol{y}$ are, however, too large to be stored in the fastest GPU memory (thread specific registers), and the cost of repeatedly accessing data from slower memory can be very large. To solve this problem, BROCCOLI instead updates $\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}$ after a change in $\boldsymbol{\rho}$, according to

$$\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}} = \sum_{i=0}^{p}\sum_{j=0}^{p}\rho_i\rho_j\boldsymbol{S}_{ij}, \tag{24}$$

where we for convenience define $\rho_0 = -1$, p is the order of the AR model and $\boldsymbol{S}_{ij} = \sum_{t=1}^{N}\boldsymbol{x}_{t-i}\boldsymbol{x}_{t-j}^T$ are data matrices independent of $\boldsymbol{\rho}$ ($\boldsymbol{x}_t$ is a vector that contains all the regressors for time point t, while $\boldsymbol{X}$ is the full design matrix). Note that $\boldsymbol{S}_{ij} = \boldsymbol{S}_{ji}$ and that all $\boldsymbol{S}_{ij}$ are symmetric. For a first order AR model, the update is given by

$$\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}} = \boldsymbol{S}_{00} - 2\rho\boldsymbol{S}_{01} + \rho^2\boldsymbol{S}_{11}. \tag{25}$$

Note how the data matrices $\boldsymbol{S}_{ij}$ are separated from $\boldsymbol{\rho}$ in the above expressions. The $\boldsymbol{S}_{ij}$ matrices are not voxel-specific and can, therefore, be pre-computed and stored in constant memory. Analogous formulas are easily derived for $\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{y}}$ (with data moments $\boldsymbol{m}_{ij} = \sum_{t=1}^{N} \boldsymbol{x}_{t-i} y_{t-j}$) and $\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{y}}$ (with data moments $g_{ij} = \sum_{t=1}^{N} y_{t-i} y_{t-j}$), both of which are needed for the Gibbs sampling. The $\boldsymbol{m}_{ij}$ and the g_{ij} values are voxel-specific, but low-dimensional and can therefore be stored in thread specific registers. Despite these optimizations, the implementation can currently only handle a small number of regressors and an AR(1) model of the residuals. The extension to more elaborate models is in principle straight forward however, and rapid advancements in GPU memory are likely to remove these limitations in the near future

The Bayesian fMRI analysis also requires random number generation to estimate the joint posterior distribution. The CURAND library can be used for this purpose for the CUDA programming language, but there exists no similar library for OpenCL. Instead, random numbers are first generated in BROCCOLI from a uniform distribution, using a voxel/thread specific seed and a modulo operation (Langdon, 2009). This is the only part of the OpenCL code that currently uses double precision. The seeds are generated on the host side, as this operation only needs to be performed once. The uniformly distributed numbers are then used to generate numbers from a normal distribution, by applying the Box-Muller transform (Box and Muller, 1958). Random numbers from an inverse Gamma distribution can finally be generated as

$$g = \frac{2B}{\sum_{i=1}^{2A} n_i^2}, \tag{26}$$

where n is a random number from a normal distribution with zero mean and unit variance, A is the shape parameter of the Gamma distribution and B is the scale parameter.

3. RESULTS

A number of freely available fMRI datasets (Biswal et al., 2010; Poldrack et al., 2013) were used to test our software, and to compare it to existing software packages. The hardware used for testing is specified in **Table 1**. Specifically, BROCCOLI was used with an Intel CPU, an Nvidia GPU and an AMD GPU, to demonstrate that the same code can run on different types of hardware. The following software packages were compared to BROCCOLI: SPM8, FSL 5.0.4 (Smith et al., 2004) (with the package Condor installed for parallel processing) and AFNI (Cox, 1996) (with OpenMP support for parallel processing). For FSL, the shell variable FSLPARALLEL was set to "condor" to measure multi-core results. For AFNI, the shell variable OMP_NUM_THREADS was set to "1" to generate processing times for single-core processing, and to "8" for multi-core processing. BROCCOLI running on a CPU automatically uses all available processor cores for all processing steps. All testing scripts can be downloaded from github [10]. To make the comparison reflective of each package's standard use, our testing scripts were posted on the mailing lists for SPM, FSL, and AFNI and modified according to responses.

It should be stressed that the different software packages use different algorithms, programming languages and libraries. It is therefore hard to make a quantitatively meaningful performance comparison. For this reason, we also added the processing time for BROCCOLI running on a single CPU core, such that there is a baseline comparison for each algorithm. This was achieved by setting the shell variable CPU_MAX_COMPUTE_UNITS to 1 (a more general and complicated way is to use OpenCL device fission).

3.1. SPATIAL NORMALIZATION

The quality of the normalization to MNI space was tested by aligning 198 T1-weighted volumes to the MNI brain templates (1 and 2 mm^3 resolution) provided in the FSL software (MNI152_T1_1 mm_brain.nii.gz, MNI152_T1_2 mm_brain.nii.gz). The T1 volumes were downloaded from the 1000 functional connectomes project (Biswal et al., 2010), and the Cambridge dataset was selected for its large number of subjects. Each T1 volume is of the size 192 × 192 × 144 voxels with a resolution of 1.2 × 1.2 × 1.2 mm. To fully focus on the registration algorithm, the provided skullstripped T1 volumes were used rather than the original T1 volumes.

For SPM the functions "Normalize" and "Segment" were used for normalization. For "Normalize," the parameter 'Source image smoothing' was changed from 8 mm to 4 mm, to try to match the smoothness of the FSL T1 template (the T1 template in SPM is more blurred than the T1 template in FSL). For 'Segment', an initial parametric alignment of each T1 volume was first performed using the function 'Coregister' (otherwise several normalized T1 volumes were far off from the MNI template). Except for these modifications, the default settings were used. For FSL, the T1 volumes were aligned by running FLIRT (which performs linear registration) using the skullstripped volume and template, followed by FNIRT (which performs non-linear registration) using the volume and template with skull (this is the recommended usage). The estimated deformation field was finally applied to the skullstripped volume. For registration to the 2 mm^3 MNI template, the configuration file "T1_2_MNI152_2 mm.cnf" was used, while the default settings were used for registration to the 1 mm^3 template (there is no "T1_2_MNI152_1 mm.cnf"). For AFNI, alignment was performed correspondingly by running 3dUnifize (which normalizes the image intensity) both for the T1 volume and the MNI template, 3dAllineate and 3dQwarp. The estimated displacement field was finally applied to the original T1 volume without intensity normalization, using the function 3dNwarpApply. The default interpolation method for 3dNwarpApply is sinc interpolation, but as SPM, FSL and BROCCOLI all use linear interpolation by default, 3dNwarpApply was tested with linear as well as sinc

[10] https://github.com/wanderine/BROCCOLI/tree/master/code/testing_scripts

interpolation. The non-linear registration in 3dQwarp is done with a combination of cubic and quintic basis functions, and it is not possible to change this to linear interpolation. Since 3dQwarp in AFNI is a very new method, we used settings proposed in the help text. For all software packages, the same settings were used for each T1 volume.

Average normalized T1 volumes were calculated for SPM, FSL, AFNI, and BROCCOLI, to visually compare the algorithms, and are given in **Figure 3**. It should be noted that for FSL, the resulting displacement field from the 2 mm^3 normalization was upscaled and used to generate the normalized T1 volumes used here (as recommended by the FSL mailing list). For a more numerical comparison of the image registration quality, the normalized cross-correlation, mutual information and sum of squared differences were calculated between each normalized T1 volume and the MNI template, the mean results are given in **Figure 4**. Only the voxels inside the MNI brain mask were used to calculate these similarity measures, as 75% of the voxels are outside the brain. The processing time for the different software packages are given in **Figure 5**.

3.2. MOTION CORRECTION

The motion correction algorithms in SPM (realign), FSL (MCFLIRT), AFNI (3dvolreg), and BROCCOLI were tested by using test datasets with known motion parameters. The test datasets were generated by repeatedly using only the first fMRI volume in each dataset and applying known random rigid transformations to this first volume. The translations and rotations were independently generated from a normal distribution with a mean of 0 and a standard deviation of 0.5 (voxels for translations and degrees for rotations). Gaussian white noise was then added to each volume. To further demonstrate the robustness of BROCCOLI's phase based algorithm, a shading was added to each transformed fMRI volume. An example of the added shading is given in **Figure 6**. The test datasets were created using the 198 resting state datasets in the Cambridge dataset (Biswal et al., 2010). Each rest dataset is of the size 72 × 72 × 47 × 119 with a voxel resolution of 3 × 3 × 3 mm.

For SPM and AFNI, the algorithms were tested with linear interpolation in addition to the default setting (b-spline for SPM and Fourier for AFNI), as FSL and BROCCOLI use linear interpolation as default. For SPM and FSL, the reference volume was set to the first volume, which is the default for AFNI and BROCCOLI. Except for these changes, the default settings were used for all software packages. The quality of the motion correction was evaluated by comparing the estimated transformations to the true ones. For each dataset, the total error was calculated as the square root of the sum of the squared differences over all motion parameters p and time points t, i.e.,

$$\epsilon = \sqrt{\sum_{t=1}^{119} \sum_{p=1}^{6} \left(motion_{estimated}(t,p) - motion_{true}(t,p) \right)^2}. \qquad (27)$$

The mean error measures for the different software packages, averaged over the 198 subjects, are given in **Figure 7** and the processing times for motion correction are given in **Figure 8**.

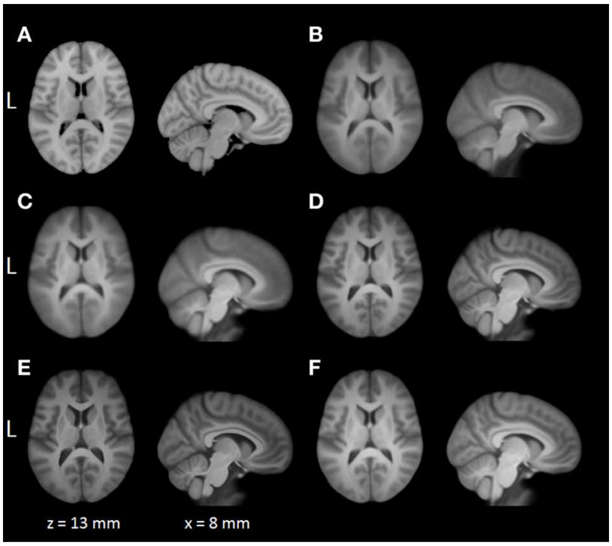


FIGURE 3 | A visual comparison of spatial normalization with the different software packages, by averaging 198 normalized T1 volumes. (A) MNI template, **(B)** SPM Normalize average normalized T1 volume, **(C)** SPM Segment average normalized T1 volume, **(D)** FSL average normalized T1 volume, **(E)** AFNI average normalized T1 volume, **(F)** BROCCOLI average normalized T1 volume. Note that AFNI uses a combination of cubic, quintic, and sinc interpolation as default, while SPM, FSL, and BROCCOLI all use linear interpolation as default.

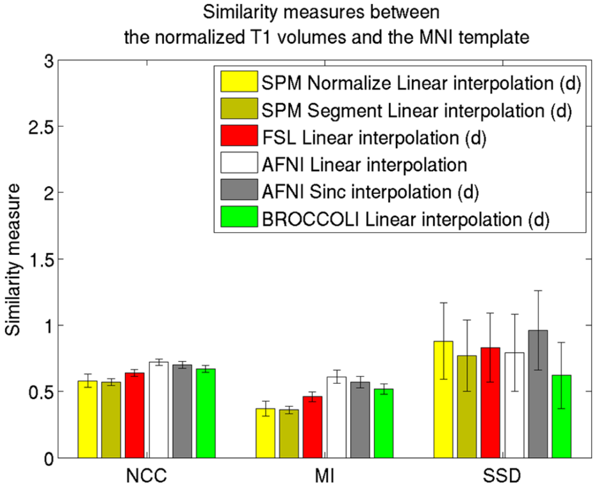


FIGURE 4 | Similarity measures between each normalized T1 volume and the MNI template for the different software packages, averaged over 198 subjects. The error bars represent the standard deviation. NCC stands for normalized cross correlation (higher is better), MI stands for mutual information (higher is better) and SSD stands for sum of squared differences (lower is better). For visualization purposes, the SSD similarity measure was divided by 300,000. Only the voxels in the MNI brain mask were used to calculate these similarity measures, as 75% of the voxels are outside the brain. Different interpolation modes were tested, as the software packages have different default settings for interpolation (d denotes the default interpolation).

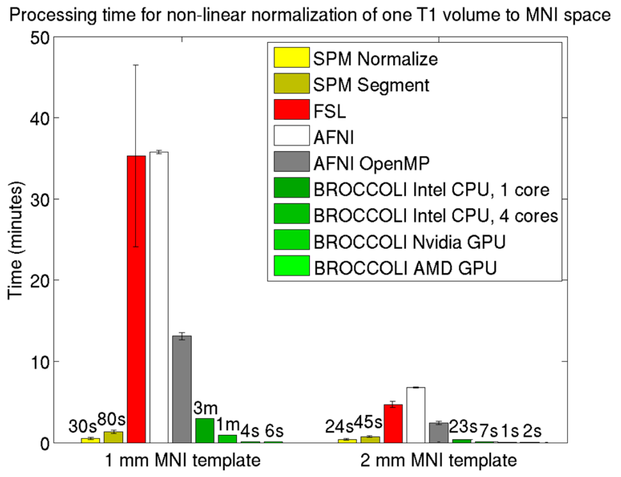


FIGURE 5 | Processing times for non-linear spatial normalization of one T1 volume of size 192 × 192 × 144 voxels to a MNI template (1 and 2 mm^3 resolution) for the different software packages, averaged over 198 T1 volumes. The error bars represent the standard deviation. Note that AFNI uses a combination of cubic, quintic, and sinc interpolation as default, while SPM, FSL, and BROCCOLI all use linear interpolation as default. A linear registration was first applied to achieve a good starting point for the non-linear registration. BROCCOLI running on a GPU can perform non-linear normalization to a 1 mm^3 template in 4–6 s, and still provide a satisfactory result. BROCCOLI running on a CPU is also significantly faster than FSL and AFNI OpenMP, even if a single CPU core is used.

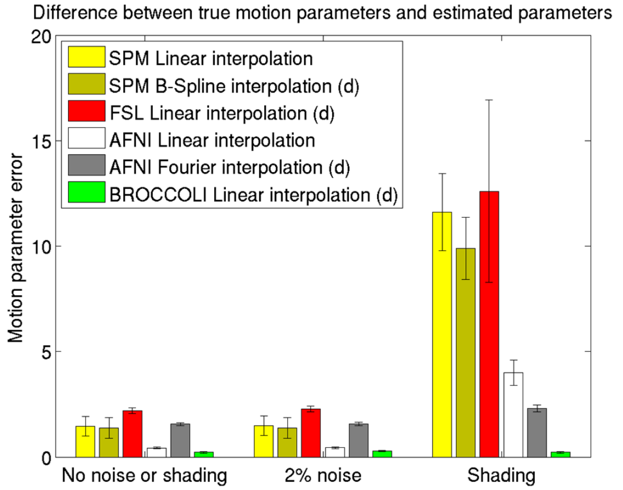


FIGURE 7 | Motion parameter errors for the different software packages, averaged over 198 datasets with artificial motion. The error bars represent the standard deviation. The testing datasets were generated by applying random translations and rotations to the first fMRI volume in each dataset, and then adding Gaussian noise or a shading. The amount of noise was defined by setting the standard deviation to a percentage of the maximum intensity value. Different interpolation modes were tested, as the software packages have different default settings for interpolation (d denotes the default interpolation). The presented results were generated with an Nvidia GPU, and equal results were also obtained by the Intel CPU and the AMD GPU.

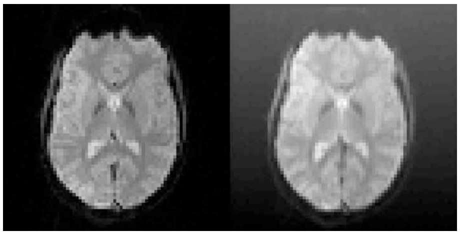


FIGURE 6 | Left: One slice of one fMRI dataset used for testing the motion correction algorithms. **Right:** The same slice after the application of a random translation and rotation, and addition of a shading (gradient) increasing upwards. The shading will affect all algorithms that use the image intensity directly. The phase based algorithm used in BROCCOLI will, however, not be affected by this shading. The main reason for this is that quadrature filters are bandpass filters, which remove low frequency variations (e.g., shadings) as well as high frequency variations (e.g., noise).

3.3. FIRST LEVEL ANALYSIS

The first level analysis was tested by analyzing freely available task fMRI datasets, downloaded from the OpenfMRI (Poldrack et al., 2013) homepage. Specifically, the OpenfMRI "rhyme judgment" dataset was used where the subjects were presented with pairs of either words or pseudowords, and made rhyming judgments for each pair. See the work by Xue and Poldrack (2007) for further information about this dataset.

3.3.1. Frequentist inference

To the best of our knowledge, the SPM software package does not have any default processing pipeline. Instead, we used a batch script for first level analysis available on the SPM homepage[11]. For FSL, the analysis was setup and started through the graphical user interface. For AFNI, the Python script afni_proc.py was used, through the graphical interface uber_subject.py. The settings used for each software are given in **Table 2**. Processing times for first level analysis for the different software packages are given in **Figure 9**. A visual comparison of one brain activity map, for BROCCOLI and FSL, is given in **Figure 10**. Processing times for BROCCOLI for a first level permutation-based analysis, using 10,000 permutations, are given in **Figure 11**.

3.3.2. Bayesian inference

The Bayesian fMRI analysis was tested by generating a total of 11,000 draws from the posterior distribution for each brain voxel (44,220 voxels), and the first 1000 draws were discarded as "burn in" samples. The PPM was calculated as the percentage of draws where the GLM weight of interest was larger than zero. The resulting PPM is given in **Figure 12**, and can be compared to the t-map in **Figure 10**. The processing time was 4706 s using the Intel CPU and one core, 835 s using the Intel CPU and all the four cores, 190 s for the Nvidia GPU and 91 s for the AMD GPU. This can be compared to about 20 h for a naive Matlab implementation.

3.4. SECOND LEVEL ANALYSIS

To test the second level analysis, the permutation functionality in BROCCOLI was compared to the function randomize

[11] http://www.fil.ion.ucl.ac.uk/spm/data/face_rep/face_rep_spm5_batch.m

in FSL (SPM and AFNI do not have any support for permutation based analysis, although AFNI for example has support for Kruskal-Wallis tests and Wilcoxon tests). The function randomize_parallel in FSL automatically divides the number of permutations to a number of computers or CPU cores (if for example Condor or GridEngine is installed), and was therefore also used for testing. First level results generated by FSL (downloaded from the OpenfMRI homepage) were used as inputs to the second level analysis, to fully focus on the permutation procedure. Here we used the OpenfMRI dataset "word and object processing", as it has the largest number of subjects (49). See the work by Duncan et al. (2009) for further information about this dataset. Processing times for FSL and BROCCOLI for a second level permutation-based analysis of the 49 subjects, using 10,000 permutations, are given in **Figure 13**. Null distributions generated by FSL and BROCCOLI, for a design matrix containing a single regressor, were compared numerically and were found to be equivalent. A direct comparison for more than one regressor is more problematic, as the randomize function in FSL first transforms the design matrix to effective regressors and effective confound regressors, by using information from the contrast vector.

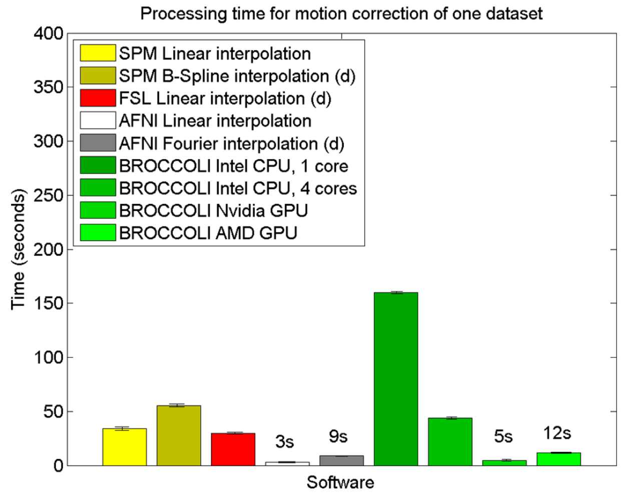


FIGURE 8 | Processing times for motion correction of one fMRI dataset of size 72 × 72 × 47 × 119 for the different software packages, averaged over 198 datasets. The error bars represent the standard deviation. All algorithms registered all volumes to the first one. The processing times for AFNI and AFNI OpenMP are the same, as the AFNI software does not have any OpenMP support for motion correction. Different interpolation modes were tested, as the software packages have different default settings for interpolation (d denotes the default interpolation).

4. DISCUSSION

We have presented a new software package for fMRI analysis. BROCCOLI is written in OpenCL, making it possible to run the analysis in parallel, taking full advantage of a large variety of hardware configurations. To exemplify this, BROCCOLI has been tested with an Intel CPU, an Nvidia GPU and an AMD GPU. The main objective of BROCCOLI is to demonstrate the advantages of parallel processing and to enable the neuroimaging field to avail itself of more computationally demanding normalization algorithms, and statistical methods that are based on a smaller number of assumptions (e.g., by using non-parametric statistics). Currently, BROCCOLI reduces the fMRI processing time by at least an order of magnitude compared to existing software packages (even if only a CPU and not a GPU is used). For non-linear spatial normalization, BROCCOLI running on an Nvidia GPU is approximately 525 times faster compared to FSL and AFNI, and 195 times faster than AFNI OpenMP. For second level permutation tests, BROCCOLI using an Nvidia GPU is 100–200 times faster than FSL and 33–130 times faster than the parallel version of FSL.

4.1. SPATIAL NORMALIZATION

The accuracy measures illustrated in **Figures 3** and **4** reveal a number of interesting differences. The normalization in AFNI yields the highest mean correlation and mutual information. It might seem non-intuitive that the sinc interpolation in AFNI gives a higher sum of squared differences compared to the linear interpolation, but this is possibly explained by the fact that the sinc interpolation preserves high resolution details, perhaps beyond the meaningful resolution of the MNI template. The average normalized T1 volumes generated by SPM are clearly the most blurred, although the algorithms are fast compared to FSL and AFNI. The results presented here are consistent with a previous comparison (Klein et al., 2009), where the FSL function FNIRT was shown to provide better normalizations than the SPM functions "Segment" and "Normalize." AFNI was not included in this comparison, as the function 3dQwarp was released recently.

These comparisons should not be considered as a thorough head-to-head evaluation of the different software packages.

Table 2 | Settings for first level analysis for the different software packages (for AFNI it is currently not possible to select non-linear registration in the graphical user interface).

	Normalization	Motion	Motion regressors	Smoothing (mm)	Cluster simulation	Modeling of GLM residuals
SPM	Linear + non-linear to MNI template	Yes	Yes, 6	6	Not available	Global AR(1)
FSL	Linear + non-linear to MNI template	Yes	Yes, 6	6	Not available	FILM prewhitening (Woolrich et al., 2001)
AFNI	Linear to MNI template	Yes	Yes, 6	6	No	Voxel-wise ARMA(1, 1)
BROCCOLI	Linear + non-linear to MNI template	Yes	Yes, 6	6	Not available	Voxel-wise AR(4)

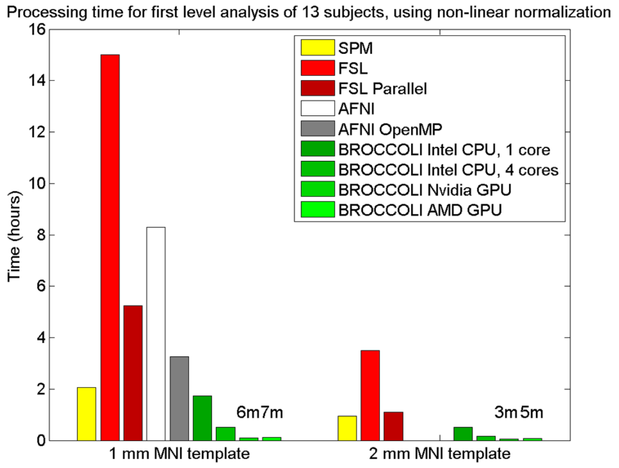


FIGURE 9 | Processing times for first level analysis of 13 fMRI datasets (of size 64 × 64 × 33 × 160). The analysis includes non-linear normalization to a brain template, slice timing correction, motion correction, smoothing, and statistical analysis. A Matlab script, available on the SPM homepage, was used for SPM. For FSL, the analysis was setup and started through the graphical user interface. For AFNI, the analysis was performed with *afni_proc.py*, through the graphical user interface *uber_subject.py*. It should be noted that SPM, FSL and BROCCOLI use linear and non-linear registration, while AFNI uses linear registration only (currently, it is not possible to select non-linear registration in uber_subject.py). To compensate for this, the non-linear registration for AFNI was done separately. Note that it is not possible to select a 2 mm^3 brain template in uber_subject.py, these processing times are therefore not defined. Also note that the processing times for BROCCOLI do not include any first level permutation test.

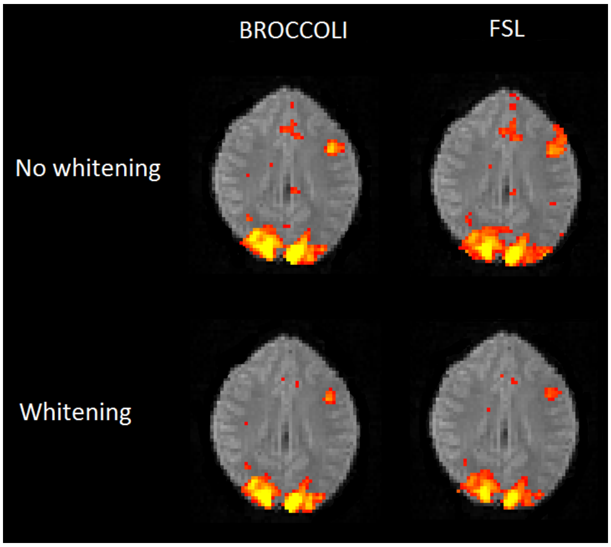


FIGURE 10 | Brain activity maps (representing *t*-values) from first level analysis of one OpenfMRI dataset, for BROCCOLI and FSL. Subjects were presented with pairs of either words or pseudowords in a block based design, and made rhyming judgments for each pair. The first level analysis here includes motion correction, segmentation of the fMRI data, smoothing, and statistical analysis. Both BROCCOLI and FSL used motion regressors in the statistical analysis. As BROCCOLI and FSL use different models of the GLM residuals, we here present activity maps with and without whitening. The activity maps have been arbitrarily thresholded at a *t*-value of 5.

Rather, the motivation was to show that BROCCOLI can provide a satisfactory normalization to MNI space in a short amount of time. An aspect not considered here, for example, is the smoothness of the resulting displacement fields. It is also possible that the different algorithms would perform better if the default settings were changed.

4.2. MOTION CORRECTION

The evaluation of the motion correction algorithms shows that BROCCOLI yields the smallest difference between the true motion parameters and the estimated ones, closely followed by AFNI. BROCCOLI using a GPU and AFNI perform the motion correction in a similar amount of time, while SPM and FSL are significantly slower. For BROCCOLI running on a CPU, the processing time is rather long, which is mainly explained by the fact that three (non-separable) quadrature filters need to be applied for each time point and for each iteration (3–5 iterations of the linear registration algorithm is normally sufficient for motion correction). BROCCOLI also estimates 12 affine registration parameters for each time point, and then restricts them to a rigid transformation (6 parameters). The results presented here are consistent with a previous comparison of motion correction algorithms (Oakes et al., 2005), where the AFNI software was shown to provide the most accurate motion estimates.

It should be noted that the test used here is not based on realistic head motion, as completely random transformations were applied for each time point. This can, for example, negatively effect the MCFLIRT function used in FSL. The reason for this is that MCFLIRT uses the motion estimate from the previous time point as a starting estimate for the next time point (Jenkinson et al., 2002). Similarly, 3dvolreg in AFNI is only intended for small motions, and the transformations applied here may have been too severe. The shading test is also not very realistic, but clearly shows the robustness of phase based registration algorithms compared to intensity based algorithms. For these reasons, the presented results should be interpreted with caution.

4.3. FIRST LEVEL ANALYSIS

4.3.1. Frequentist inference

The first level analysis using FSL and BROCCOLI yield very similar results, both with and without pre-whitening to correct for auto correlation in the GLM residuals. The small differences in activation between FSL and BROCCOLI can be explained by a number of factors. The motion correction algorithms, for example, provide slightly different results according to **Figure 7** and this will affect further processing. There are also some differences in how FSL and BROCCOLI setup the design matrix and treat the auto correlation of the GLM residuals. BROCCOLI uses four detrending regressors (mean, linear trend, quadratic trend, cubic trend) while FSL instead applies a temporal filtering to the data and the regressors. BROCCOLI smooths all the AR estimates in the same way, while FSL separately smooths AR estimates in white and gray brain matter (Woolrich et al., 2001).

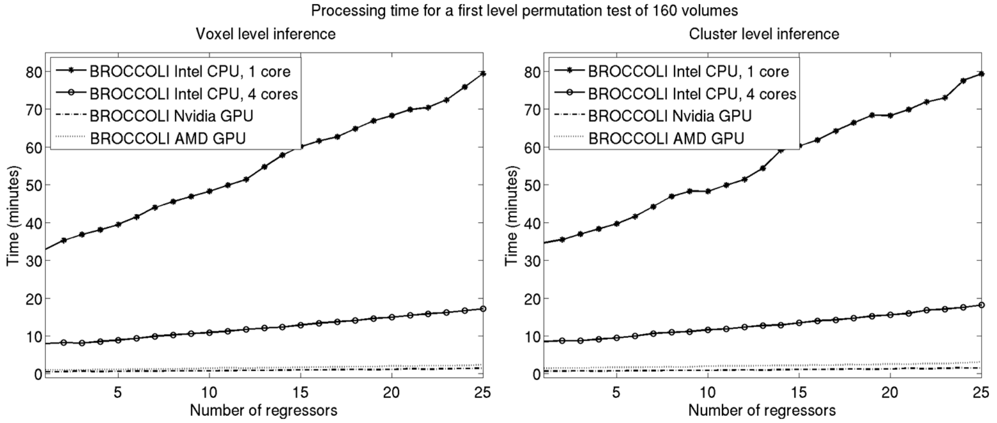


FIGURE 11 | Processing times for BROCCOLI for first level analysis using a permutation based *t*-test with 10,000 permutations (SPM, FSL, and AFNI do not provide any functions for first level permutation based analysis). Left: Voxel-level inference, the maximum *t*-test value is saved in each permutation. **Right:** Cluster-level inference, the extent of the largest cluster is saved in each permutation. A *t*-value of 3 was used as a cluster defining threshold. The data used is of the size 64 × 64 × 33 × 160. A brain mask was used to only perform the statistical calculations for the brain voxels. Note that these processing times do not include smoothing in each permutation. Smoothing the fMRI data 10,000 times takes about 8970 s using one core on the Intel CPU, 2710 s using all the four cores on the Intel CPU, 335 s with the Nvidia GPU and 550 s with the AMD GPU. Also note that ordinary least squares is used to estimate the GLM beta weights in each permutation, and not the more demanding Cochrane-Orcutt procedure.

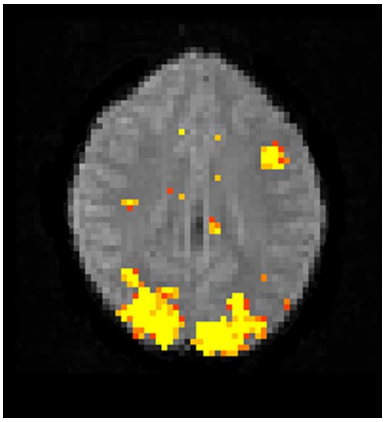


FIGURE 12 | A posterior probability map (PPM) from a Bayesian first level analysis of one OpenfMRI dataset. Subjects were presented with pairs of either words or pseudowords in a block based design, and made rhyming judgments for each pair. The first level analysis here includes motion correction, segmentation of the fMRI data, smoothing, and statistical analysis. The PPM represents the probability of the first GLM beta weight being larger than zero, and has been arbitrarily thresholded at a probability of 0.99. Note that the PPM has been calculated by using a Gibbs sampler, and not by using techniques based on variational Bayes. Also note that the frequentist approach uses a voxel-wise AR(4) model of the GLM residuals, while the Bayesian currently uses a voxel-wise AR(1) model (due to hardware limitations).

BROCCOLI is significantly faster than SPM, FSL, and AFNI, even when the analysis is run on a CPU. SPM is also faster than FSL and AFNI, which is mainly explained by a faster spatial normalization. The parallel version of FSL, where one first level analysis in our case runs on each CPU thread, is significantly faster than the non-parallel version. However, as the first level analysis in FSL requires more than 2 GB of memory, we were only able to run 6 (instead of 8) threads in parallel (since the computer used for testing has 16 GB of memory).

4.3.2. *Bayesian inference*

The Bayesian first level analysis yields results that are similar to the *t*-maps, although the results cannot be compared directly. It might seem confusing that the AMD GPU is faster than the Nvidia GPU, especially since the Nvidia GPU is faster for permutation tests. The reason for this is that the random number generation currently uses double precision, and the AMD GPU used in our case has better support for such calculations than the Nvidia GPU (see **Table 1**).

4.4. SECOND LEVEL ANALYSIS

The processing times in **Figure 13** for the second level permutation test may at first appear confusing. The speedup of using randomize_parallel instead of randomize decreases with the number of regressors, from a speedup of 3.2 for a single regressor to 1.6 for 25 regressors (but the actual time saved increases). The 10,000 permutations are divided into smaller work items of 300 permutations each for randomize_parallel. However, 33 work items cannot be divided equally to a CPU running 8 threads (8 threads $*$ 4 work items per thread $=$ 32 work items). The permutation test is therefore not completed until the last work item has been processed, for which only a single CPU thread is active. The unequal division is more problematic for more regressors, as each work item then takes a longer time to process.

The processing time for BROCCOLI is not affected as much by the number of GLM regressors as the FSL software is, resulting in a larger speedup for a larger number of regressors. A GPU thread that performs a small number of calculations is very limited by the memory bandwidth. More regressors lead to more calculations and, thereby, a better utilization of the

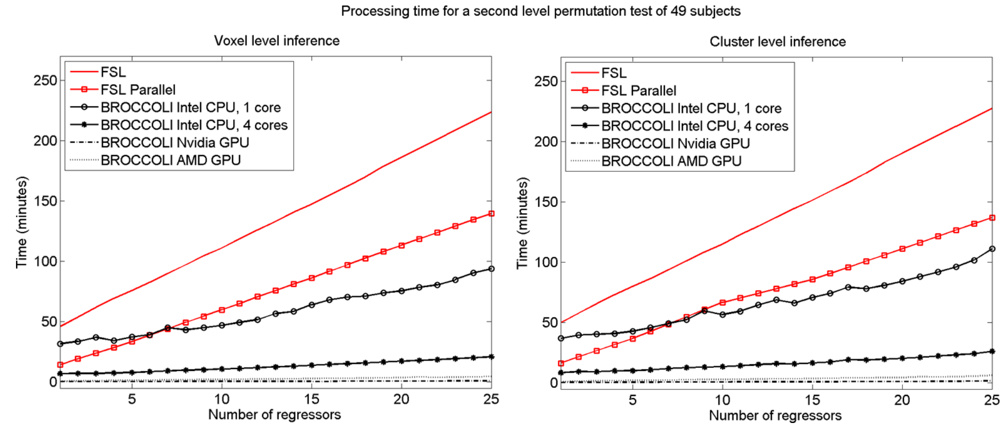


FIGURE 13 | Processing times for second level analysis using a permutation based *t*-test with 10,000 permutations, for BROCCOLI and FSL (the SPM and AFNI software packages do not provide any functions for permutation based analysis). Left: Voxel-level inference, the maximum *t*-test value is saved in each permutation. **Right:** Cluster-level inference, the extent of the largest cluster is saved in each permutation. A *t*-value of 3 was used as a cluster defining threshold. The data used are beta volumes from 49 subjects, where each beta volume is of the size 91 × 109 × 91 voxels. A brain mask was used to only perform the statistical calculations for the brain voxels. The processing time for FSL increases quickly with the number of regressors, while the processing time for BROCCOLI increases much more slowly. This is explained by the fact that calculations on a GPU are efficient, once all the data have been loaded from the slow global memory to the fast thread specific registers. To estimate several beta weights per voxel, instead of a single weight, therefore results in a better utilization of the GPU performance. The processing time for BROCCOLI using an AMD GPU is 2–5 times as high compared to BROCCOLI using a Nvidia GPU. One possible explanation for this is that the code was converted from CUDA to OpenCL. Note that these processing times are for data normalized to a 2 mm^3 MNI template. The permutation tests would take approximately eight times longer for data normalized to a 1 mm^3 MNI template.

computational capabilities of a GPU. BROCCOLI running on a CPU is also faster than the parallel version of FSL. FSL divides the work into several CPU cores by using a package like Condor or GridEngine. Such an approach cannot as easily take advantage of vectorized operations [e.g., Intel streaming SIMD extensions (SSE)], where the same operation is applied to a number of elements simultaneously. Note that this is a distinct, second layer of parallel processing. In addition to the code running on several CPU cores instead of just one, the processing on each individual core is vectorized, performing 4–16 arithmetic operations on different data at once.

It should also be noted that the presented processing times are for fMRI data registered to a 2 mm^3 MNI template, each permutation test would take approximately 8 times longer for data registered to a 1 mm^3 MNI template. Threshold free cluster enhancement (Smith and Nichols, 2009) is another inference method that would benefit from GPU acceleration, as it is much more computationally demanding compared to voxel-level and cluster-level inference.

4.5. LIMITATIONS

The following list itemizes the current limitations of using BROCCOLI:

- BROCCOLI currently has very limited support for image segmentation, but such algorithms are often easy to run in parallel (Eklund et al., 2013a).
- The quality of the fMRI-to-T1 registration has not been tested as extensively as the T1-to-MNI registration. There are, at least, two reasons why the fMRI-to-T1 registration is harder to test than the T1-to-MNI registration. First, the fMRI data is of much lower spatial resolution and an average of 198 registered fMRI volumes would therefore be extremely blurry. Second, the fMRI data is often distorted due to artifacts from the MRI sequence.
- The SPM, FSL, and AFNI software packages have been used for a long time and have been extensively tested, while BROCCOLI is completely new software.
- SPM, FSL, and AFNI all provide a graphical user interface, which BROCCOLI currently does not.
- SPM, FSL, and AFNI all provide a large number of functions which can be combined to basically solve any problem. BROCCOLI is on the other hand currently limited to image registration and first and second level fMRI analyses.
- SPM, FSL, and AFNI all provide some sort of community forum where users can get help.

4.6. FUTURE WORK

In the future, BROCCOLI can be improved and extended in several ways. The most important addition may be a graphical user interface, so that as many researchers as possible can take advantage of parallel processing. For the first version of BROCCOLI we have focused on functionality and stability, and not so much on the computational performance. As most of the code was converted from CUDA to OpenCL, it is likely that BROCCOLI performs best for Nvidia GPUs. Optimizing the code for other hardware platforms (e.g., Intel and AMD) will therefore be one important project (Enmyren and Kessler, 2010). For permutation tests involving large datasets, multi-GPU support can be used to further reduce the computational burden, by running a number of permutations on each GPU (Eklund et al., 2011a). First level analysis can also run in parallel on several GPUs with multi-GPU support, such that each GPU independently processes one subject. Another natural extension would be to provide several other wrappers for BROCCOLI, such as R and bash.

Rather than using ordinary least squares to estimate beta weights in the GLM, it would be interesting to, for example, use a regularized regression approach such as LASSO (Tibshirani, 1996) instead. LASSO is often used together with cross validation, and would be rather time consuming to run for every voxel. This is especially true if LASSO is combined with a permutation procedure, to correct for multiple comparisons. Most fMRI researchers use the GLM for the statistical analysis, but multivariate approaches that adaptively combine timeseries of several voxels can, in some cases, yield higher statistical power. We would therefore also like to convert our existing CUDA code for canonical correlation analysis (CCA) (Friman et al., 2003; Eklund et al., 2011a) to OpenCL and include it in BROCCOLI. The null distribution of canonical correlations is much more complicated than conventional t-tests, a problem which can be solved with permutation-based procedures.

ACKNOWLEDGMENTS

The authors would like to thank Daniel Forsberg for providing valuable guidance with the Morphon registration algorithm, Cyrus Eierud for beta testing the software and providing useful feedback, the FSL developer Jesper Andersson for helping us with the non-linear registration in FSL, Morgan Hough and Miha Čančula for initiating and developing the Python wrapper and Daniel Playne for helping us with the parallel clustering algorithm. The authors also thank the 1000 functional connectomes project and the OpenfMRI project for sharing data.

REFERENCES

Ben-Shalom, R., Liberman, G., and Korngreen, A. (2013). Accelerating compartmental modeling on a graphical processing unit. *Front. Neuroinform.* 7:4. doi: 10.3389/fninf.2013.00004

Biswal, B., Mennes, M., Zuo, X.-N., Gohel, S., Kelly, C., Smith, S. M., et al. (2010). Toward discovery science of human brain function. *Proc. Natl. Acad. Sci. U.S.A.* 107, 4734–4739. doi: 10.1073/pnas.0911855107

Box, G., and Muller, M. E. (1958). A note on the generation of random normal deviates. *Ann. Math. Stat.* 29, 610–611. doi: 10.1214/aoms/1177706645

Button, K., Ioannidis, J., Mokrysz, C., Nosek, B., Flint, J., Robinson, E., et al. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nat. Rev. Neurosci.* 14, 365–376. doi: 10.1038/nrn3475

Cochrane, D., and Orcutt, G. (1949). Application of least squares regression to relationships containing autocorrelated error terms. *J. Am. Stat. Assoc.* 44, 32–61.

Cox, R. (1996). AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput. Biomed. Res.* 29, 162–173. doi: 10.1006/cbmr.1996.0014

da Silva, A. R. F. (2011). cudaBayesreg: parallel implementation of a Bayesian multilevel model for fMRI data analysis. *J. Stat. Softw.* 44, 1–24. Available online at: http://www.jstatsoft.org/v44/i04

Day, M. (2012). *Extracting Euler Angles from A Rotation Matrix. Insomniac Games R&D*. Available online at: http://www.insomniacgames.com/mike-day-extracting-euler-angles-from-a-rotation-matrix/

Duncan, K., Pattamadilok, C., Knierim, I., and Devlin, J. (2009). Consistency and variability in functional localisers. *Neuroimage* 46, 1018–1026. doi: 10.1016/j.neuroimage.2009.03.014

Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *Ann. Math. Stat.* 28, 181–187. doi: 10.1214/aoms/1177707045

Eklund, A., Andersson, M., Josephson, C., Johannesson, M., and Knutsson, H. (2012a). Does parametric fMRI analysis with SPM yield valid results? - an empirical study of 1484 rest datasets. *Neuroimage* 61, 565–578. doi: 10.1016/j.neuroimage.2012.03.093

Eklund, A., Andersson, M., and Knutsson, H. (2012b). fMRI analysis on the GPU - Possibilities and challenges. *Comput. Methods Prog. Biomed.* 105, 145–161. doi: 10.1016/j.cmpb.2011.07.007

Eklund, A., Andersson, M., and Knutsson, H. (2010). "Phase based volume registration using CUDA," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Dallas, TX), 658–661. doi: 10.1109/ICASSP.2010.5495134

Eklund, A., Andersson, M., and Knutsson, H. (2011a). Fast random permutation tests enable objective evaluation of methods for single subject fMRI analysis. *Int. J. Biomed. Imaging* 2011:627947. doi: 10.1155/2011/627947

Eklund, A., Forsberg, D., Andersson, M., and Knutsson, H. (2011b). "Using the local phase of the magnitude of the local structure tensor for image registration," in *Lecture Notes in Computer Science, Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, (Ystad), 6688, 414–423. doi: 10.1007/978-3-642-21227-7 39

Eklund, A., and Dufort, P. (2014). "Non-separable 2D, 3D and 4D filtering with CUDA," in *GPU Pro 5*, ed W. Engel (Natick, MA: A K Peters/CRC Press), 465–487.

Eklund, A., Dufort, P., Forsberg, D., and LaConte, S. (2013a). Medical image processing on the GPU - Past, present and future. *Med. Image Anal.* 17, 1073–1094. doi: 10.1016/j.media.2013.05.008

Eklund, A., Villani, M., and LaConte, S. (2013b). Harnessing graphics processing units for improved neuroimaging statistics. *Cogn. Affect. Behav. Neurosci.* 13, 587–597. doi: 10.3758/s13415-013-0165-7

Enmyren, J., and Kessler, C. (2010). "SkePU: a multi-backend skeleton programming library for multi-GPU systems," in *Proceedings of International Workshop on High-level Parallel Programming and Applications (HLPP)* (New York, NY: ACM), 5–14. doi: 10.1145/1863482.1863487

Feinberg, D., and Yacoub, E. (2012). The rapid development of high speed, resolution and precision in fMRI. *Neuro mage* 62, 720–725. doi: 10.1016/j.neuroimage.2012.01.049

Fleet, D., and Jepson, A. (1990). Computation of component image velocity from local phase information. *Int. J. Comput. Vis.* 5, 77–104. doi: 10.1007/BF00056772

Fluck, O., Vetter, C., Wein, W., Kamen, A., Preim, B., and Westermann, R. (2011). A survey of medical image registration on graphics hardware. *Comput. Methods Prog. Biomed.* 104, e45–e57. doi: 10.1016/j.cmpb.2010.10.009

Forsberg, D. (2013). *Robust Image Registration for Improved Clinical Efficiency: Using Local Structure Analysis and Model-Based Processing. Linköping Studies in Science and Technology.* Ph.D. thesis No. 1514, Linköping University, Sweden.

Forsberg, D., Eklund, A., Andersson, M., and Knutsson, H. (2011). "Phase-based non-rigid 3D image registration - from minutes to seconds using CUDA," in *Joint MICCAI Workshop on High Performance and Distributed Computing for Medical Imaging* (Toronto).

Friman, O., Borga, M., Lundberg, P., and Knutsson, H. (2003). Adaptive analysis of fMRI data. *Neuroimage* 19, 837–845. doi: 10.1016/S1053-8119(03)00077-6

Friston, K., Holmes, A., Worsley, K., Poline, J., Frith, C., and Frackowiak, R. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Hum. Brain Mapp.* 2, 189–210. doi: 10.1002/hbm.460020402

Friston, K., Penny, W., Phillips, C., Kiebel, S., Hinton, G., and Ashburner, J. (2002). Classical and Bayesian inference in neuroimaging: Theory. *Neuroimage* 16, 465–483. doi: 10.1006/nimg.2002.1090

Gautama, T., and Hulle, M. V. (2004). Optimal spatial regularisation of autocorrelation estimates in fMRI analysis. *Neuroimage* 23, 1203–1216. doi: 10.1016/j.neuroimage.2004.07.048

Genovese, C. R., Lazar, N. A., and Nichols, T. (2002). Thresholding of statistical maps in functional neuroimaging using the false discovery rate. *Neuroimage* 15, 870–878. doi: 10.1006/nimg.2001.1037

Granlund, G., and Knutsson, H. (1995). *Signal Processing for Computer Vision*. Dordrecht: Kluwer Academic Publishers. ISBN: 0-7923-9530-1. doi: 10.1007/978-1-4757-2377-9

Guo, G. (2012). Parallel statistical computing for statistical inference. *J. Stat. Theory Pract.* 6, 536–565. doi: 10.1080/15598608.2012.695705

Hanke, M., and Halchenko, Y. (2011). Neuroscience runs on GNU/Linux. *Front. Neuroinform.* 5:8. doi: 10.3389/fninf.2011.00008

Hawick, K., Leist, A., and Playne, D. (2010). Parallel graph component labelling with GPUs and CUDA. *Parallel Comput.* 36, 655–678. doi: 10.1016/j.parco.2010.07.002

Hemmendorff, M., Andersson, M., Kronander, T., and Knutsson, H. (2002). Phase-based multidimensional volume registration. *IEEE Trans. Med. Imaging* 21, 1536–1543. doi: 10.1109/TMI.2002.806581

Hernandez, M., Guerrero, G., Cecilia, J., Garcia, J., Inuggi, A., Jbabdi, S., et al. (2013). Accelerating fibre orientation estimation from diffusion weighted magnetic resonance imaging using GPUs. *PLoS ONE* 8:e61892. doi: 10.1371/journal.pone.0061892

Hoang, R. V., Tanna, D., Jayet Bray, L. C., Dascalu, S. M., and Harris, F. C. (2013). A novel CPU/GPU simulation environment for large-scale biologically-realistic neural modeling. *Front. Neuroinform.* 7:19. doi: 10.3389/fninf.2013.00019

Horn, B., and Schunck, B. (1981). Determining optical flow. *Artif. Intell.* 17, 185–203. doi: 10.1016/0004-3702(81)90024-2

Huang, T.-Y., Tang, Y.-W., and Ju, S.-Y. (2011). Accelerating image registration of MRI by GPU-based parallel computation. *Magn. Reson. Imaging* 29, 712–716. doi: 10.1371/journal.pone.0061892

Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *Neuroimage* 17, 825–841. doi: 10.1006/nimg.2002.1132

Jeong, W.-K., Beyer, J., Hadwiger, M., Blue, R., Law, C., Vazquez, A., et al. (2010). Ssecrett and neurotrace: interactive visualization and analysis tools for large-scale neuroscience data sets. *IEEE Comput. Graph. Appl.* 30, 58–70. doi: 10.1109/MCG.2010.56

Johnstone, T., Walsh, K. S. O., Greischar, L., Alexander, A., Fox, A., Davidson, R., et al. (2006). Motion correction and the use of motion covariates in multiple-subject fMRI analysis. *Hum. Brain Mapp.* 27, 779–788. doi: 10.1002/hbm.20219

Kirk, D., and Hwu, W. (2010). *Programming Massively Parallel Processors: A Hands-on Approach*. San Francisco, CA: Morgan Kaufmann. ISBN: 978-0-12-381472-2.

Klein, A., Andersson, J., Ardekani, B., Ashburner, J., Avants, B., Chiang, M.-C., et al. (2009). Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* 46, 786–802. doi: 10.1016/j.neuroimage.2008.12.037

Knutsson, H. (1989). "Representing local structure using tensors," in *Scandinavian Conference on Image Analysis (SCIA)* (Oulu), 244–251.

Knutsson, H., and Andersson, M. (2003). "What's so good about quadrature filters?," in *International Conference on Image Processing (ICIP)* (Barcelona), 61–64. doi: 10.1109/ICIP.2003.1247181

Knutsson, H., and Andersson, M. (2005). "Morphons: segmentation using elastic canvas and paint on priors," in *IEEE International Conference on Image Processing (ICIP)* (Genoa), 1226–1229. doi: 10.1109/ICIP.2005.1530283

Knutsson, H., Andersson, M., and Wiklund, J. (1999). "Advanced filter design," in *Scandinavian Conference on Image Analysis (SCIA)* (Kangerlussuaq: International Association for Pattern Recognition), 185–193.

Knutsson, H., and Westin, C.-F. (1993). "Normalized and differential convolution: methods for interpolation and filtering of incomplete and uncertain data," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (New York, NY), 515–523. doi: 10.1109/CVPR.1993.341081

Knutsson, H., Westin, C.-F., and Andersson, M. (2011). "Representing local structure using tensors II," *Lecture Notes in Computer Science, Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, (Ystad), 6688, 545–556. doi: 10.1007/978-3-642-21227-7 51

Langdon, W. (2009). "A fast high quality pseudo random number generator for nvidia CUDA," in *Proceedings of the Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers (GECCO)* (New York, NY: ACM), 2511–2514. doi: 10.1145/1570256.1570353

Lavoie-Courchesne, S., Rioux, P., Chouinard-Decorte, F., Sherif, T., Rousseau, M.-E., Das, S., et al. (2012). Integration of a neuroimaging processing pipeline into a pan-canadian computing grid. *J. Phys. Conf.* 341:012032. doi: 10.1088/1742-6596/341/1/012032

Lee, A., Yau, C., Giles, M., Doucet, A., and Holmes, C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *J. Comput. Graph. Stat.* 19, 769–789. doi: 10.1198/jcgs.2010.10039

Mellor, M., and Brady, M. (2005). Phase mutual information as similarity measure for registration. *Med. Image Anal.* 9, 330–343. doi: 10.1016/j.media.2005.01.002

Moeller, S., Yacoub, E., Olman, C., Auerbach, E., Strupp, J., Harel, N., et al. (2010). Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI. *Magn. Reson. Med.* 63, 1144–1153. doi: 10.1002/mrm.22361

Munshi, A., Gaster, B., Mattson, T., Fung, J., and Ginsburg, D. (2011). *OpenCL Programming Guide*. (Addison-Wesley Professional). ISBN: 978-0321749642.

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press. ISBN: 0262018020 9780262018029.

Nichols, T. E., and Holmes, A. P. (2002). Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Hum. Brain Mapp.* 15, 1–25. doi: 10.1002/hbm.1058

Oakes, T., Johnstone, T., Walsh, K. O., Greischar, L., Alexander, A., Fox, A., et al. (2005). Comparison of fMRI motion correction software tools. *Neuroimage* 28, 529–543. doi: 10.1016/j.neuroimage.2005.05.058

Owens, J., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A., et al. (2007). A survey of general-purpose computation on graphics hardware. *Comput. Graph. Forum* 26, 80–113. doi: 10.1111/j.1467-8659.2007.01012.x

Penny, W., Kiebel, S., and Friston, K. (2003). Variational Bayesian inference for fMRI time series. *Neuroimage* 19, 727–741. doi: 10.1016/S1053-8119(03)00071-5

Pezoa, J., Fasoli, D., and Faugeras, O. (2012). Three applications of GPU computing in neuroscience. *Comput. Sci. Eng.* 14, 40–47. doi: 10.1109/MCSE.2011.119

Poldrack, R., Barch, D., Mitchell, J., Wager, T., Wagner, A., Devlin, J., et al. (2013). Toward open sharing of task-based fMRI data: the OpenfMRI project. *Front. Neuroinform.* 7:12. doi: 10.3389/fninf.2013.00012

Pratx, G., and Xing, L. (2011). GPU computing in medical physics: a review. *Med. Phys.* 38, 2685–2697. doi: 10.1118/1.3578605

Sanders, J., and Kandrot, E. (2011). *CUDA by example - An introduction to general-purpose GPU programming*. (Addison-Wesley). ISBN: 978-0-13-138768-3.

Shams, R., Sadeghi, P., Kennedy, R. A., and Hartley, R. I. (2010). A survey of medical image registration on multicore and the GPU. *IEEE Signal Process. Mag.* 27, 50–60. doi: 10.1109/MSP.2009.935387

Shoemake, K. (1994). "Euler angle conversion," in *Graphics Gems IV*, ed P. S. Heckbert (AP Professional), 222–229. doi: 10.1016/B978-0-12-336156-1.50030-6

Sladky, R., Friston, K., Tröstl, J., Cunnington, R., Moser, E., and Windischberger, C. (2011). Slice-timing effects and their correction in functional MRI. *Neuroimage* 58, 588–594. doi: 10.1016/j.neuroimage.2011.06.078

Smith, S., and Nichols, T. (2009). Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. *Neuroimage* 44, 83–98. doi: 10.1016/j.neuroimage.2008.03.061

Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E., Johansen-Berg, H., et al. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *Math. Brain Imaging* 23, 208–219. doi: 10.1016/j.neuroimage. 2004.07.051

Suchard, M., Wang, Q., Chan, C., Frelinger, J., Cron, A., and West, M. (2010). Understanding GPU programming for statistical computation: studies in massively parallel massive mixtures. *J. Comput. Graph. Stat.* 19, 419–438. doi: 10.1198/jcgs.2010.10016

Thyreau, B., Schwartz, Y., Thirion, B., Frouin, V., Loth, E., Vollstädt-Klein, S., et al. (2012). Very large fMRI study using the IMAGEN database: sensitivity-specificity and population effect modeling in relation to the underlying anatomy. *Neuroimage* 61, 295–303. doi: 10.1016/j.neuroimage.2012.02.083

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B* 58, 267–288.

van Essen, D., Smith, S., Barch, D., Behrens, T., Yacoub, E., and for the WU-Minn HCP Consortium, K. U. (2013). The WU-Minn human connectome project: an overview. *Neuroimage* 80, 62–79. doi: 10.1016/j.neuroimage.2013.05.041

Woolrich, M., Ripley, B., Brady, M., and Smith, S. (2001). Temporal autocorrelation in univariate linear modeling of fMRI data. *Neuroimage* 14, 1370–1386. doi: 10.1006/nimg.2001.0931

Woolrich, M. W., Jenkinson, M., Brady, M., and Smith, S. M. (2004). Fully Bayesian spatio-temporal modeling of fMRI data. *IEEE Trans. Med. Imaging* 23, 213–231. doi: 10.1109/TMI.2003.823065

Worsley, K., Evans, A., Marrett, S., and Neelin, P. (1992). A three dimensional statistical analysis for CBF activation studies in the human brain. *J. Cereb. Blood Flow Metab.* 12, 900–918. doi: 10.1038/jcbfm.1992.127

Worsley, K., Liao, C., Aston, J., Petre, V., Duncan, G., Morales, F., et al. (2002). A general statistical analysis for fMRI data. *Neuroimage* 15, 1–15. doi: 10.1006/nimg.2001.0933

Xue, G., and Poldrack, R. A. (2007). The neural substrates of visual perceptual learning of words: implications for the visual word form area hypothesis. *J. Cogn. Neurosci.* 19, 1643–1655. doi: 10.1162/jocn.2007.19.10.1643

Yamazaki, T., and Igarashi, J. (2013). Realtime cerebellum: a large-scale spiking network model of the cerebellum that runs in realtime using a graphics processing unit. *Neural Netw.* 47, 103–111. doi: 10.1016/j.neunet.2013.01.019

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 09 December 2013; accepted: 24 February 2014; published online: 14 March 2014.

Citation: Eklund A, Dufort P, Villani M and LaConte S (2014) BROCCOLI: Software for fast fMRI analysis on many-core CPUs and GPUs. Front. Neuroinform. ***8****:24. doi: 10.3389/fninf.2014.00024*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 30 July 2014
doi: 10.3389/fninf.2014.00067

Pydpiper: a flexible toolkit for constructing novel registration pipelines

Miriam Friedel[1]*, Matthijs C. van Eede[1], Jon Pipitone[2], M. Mallar Chakravarty[2,3,4] and Jason P. Lerch[1,5]

[1] Mouse Imaging Centre, Hospital for Sick Children, Toronto, ON, Canada
[2] Kimel Family Translational Imaging-Genetics Research Laboratory, Research Imaging Centre, Centre for Addiction and Mental Health, Toronto, ON, Canada
[3] Department of Psychiatry, Institute of Biomaterials and Biomedical Engineering, University of Toronto, Toronto, ON, Canada
[4] Rotman Research Institute, Toronto, ON, Canada
[5] Department of Medical Biophysics, University of Toronto, Toronto, ON, Canada

Edited by:
John Van Horn, University of California, Los Angeles, USA

Reviewed by:
Shantanu H. Joshi, University of California, Los Angeles, USA
Satrajit S. Ghosh, Massachusetts Institute of Technology, USA

****Correspondence:***
Miriam Friedel, Mouse Imaging Centre, Hospital for Sick Children, 25 Orde Street, Toronto, ON M5T 3H7, Canada
e-mail: mfriedel@mouseimaging.ca

Using neuroimaging technologies to elucidate the relationship between genotype and phenotype and brain and behavior will be a key contribution to biomedical research in the twenty-first century. Among the many methods for analyzing neuroimaging data, image registration deserves particular attention due to its wide range of applications. Finding strategies to register together many images and analyze the differences between them can be a challenge, particularly given that different experimental designs require different registration strategies. Moreover, writing software that can handle different types of image registration pipelines in a flexible, reusable and extensible way can be challenging. In response to this challenge, we have created Pydpiper, a neuroimaging registration toolkit written in Python. Pydpiper is an open-source, freely available software package that provides multiple modules for various image registration applications. Pydpiper offers five key innovations. Specifically: (1) a robust file handling class that allows access to outputs from all stages of registration at any point in the pipeline; (2) the ability of the framework to eliminate duplicate stages; (3) reusable, easy to subclass modules; (4) a development toolkit written for non-developers; (5) four complete applications that run complex image registration pipelines "out-of-the-box." In this paper, we will discuss both the general Pydpiper framework and the various ways in which component modules can be pieced together to easily create new registration pipelines. This will include a discussion of the core principles motivating code development and a comparison of Pydpiper with other available toolkits. We also provide a comprehensive, line-by-line example to orient users with limited programming knowledge and highlight some of the most useful features of Pydpiper. In addition, we will present the four current applications of the code.

Keywords: neuroimaging, pipeline, image registration, software, Python

1. INTRODUCTION

Understanding the relationship between genotype and phenotype and brain and behavior is a core biomedical research challenge in the twenty-first century (Henkelman, 2010; Paus, 2010). Key recent developments have relied on three-dimensional neuroimaging in humans and animal models to aid in this endeavor. Part of the challenge of using neuroimaging to provide insight into neuroscience questions is quantitatively assessing large amounts of data in an automated, accurate and high throughput manner. Typically, a single study will produce anywhere from twenty to hundreds of images, where the end goal is the assessment of differences in neuroanatomy due to factors such as genotype, behavioral training, environment and disease.

Multipe algorithms have been developed for the analysis of neuroimaging data, ranging from tissue classification (Zijdenbos et al., 2002) to computational geometry (Fischl and Dale, 2000; Macdonald, 2000; Kim et al., 2005) to image registration and automatic segmentation (Collins et al., 1995; Heckemann et al., 2006; Chakravarty et al., 2013) or combinations thereof (Ashburner and Friston, 2000; Good et al., 2001). Image registration in particular will be the primary focus of this work, given its wide range of applications in humans (Gogtay et al., 2004; Joshi et al., 2007, 2012; Hyde et al., 2009; Klein et al., 2009; Durrleman et al., 2013) and animal models (Spring et al., 2007; Lau et al., 2008; Lerch et al., 2008; Maheswaran et al., 2009; Ellegood et al., 2013). Image registration determines the transformation mapping one image into the space of another, where the difference between these two images is thus encoded in that transformation. The analysis of those transformations, termed alternately Deformation Based Morphometry (DBM) or Tensor Based Morphometry (TBM), then produces global and local measures of changes in volume, position, and shape (Chung et al., 2001; Lepore et al., 2006).

Given that neuroimaging studies consist of more than just two images, strategies are needed to analyze entire datasets to identify shape or volume differences and provide a common space for performing analyses. There are a number of such image registration paradigms currently in use. One common approach is to align all images in a study to a common coordinate system, such as Talairach or MNI space (Evans et al., 2012). Alternatively, additional power to identify shape differences can be gained when all subjects in a study are aligned toward a single template that is representative of the population being studied (Mazziotta et al., 2001; Fonov et al., 2011). In the event that such a template does not exist, a study-specific template can be created from all subjects in the study (Guimond et al., 2000). One way to do this is through iterative, group-wise registration. In this procedure, all scans are aligned to a common target, then resampled with the resulting transforms into the target space. These resampled images are then averaged, creating a target for a subsequent alignment (Kovačević et al., 2005). The final average is then used as common space from which to analyze shape differences in the population.

The image registration processes described above are extremely effective when sufficient homology between all subjects in the study exist so that they can be registered to a common coordinate system. However, there are experiments where this is not possible (see **Figure 1**). This can be particularly true for longitudinal studies, where the same subject is scanned at multiple time points. In the case of early brain growth (Studholme, 2011; Szulc et al., 2013) or the growth of a tumor (Gazdzinski and Nieman, 2014), the anatomy of the brain changes to such an extent that insufficient homology exists to accurately register early time points to late ones. In spite of these difficulties, it is often possible to accurately register adjacent time points together if the time-series was densely sampled (Lerch et al., Manuscript

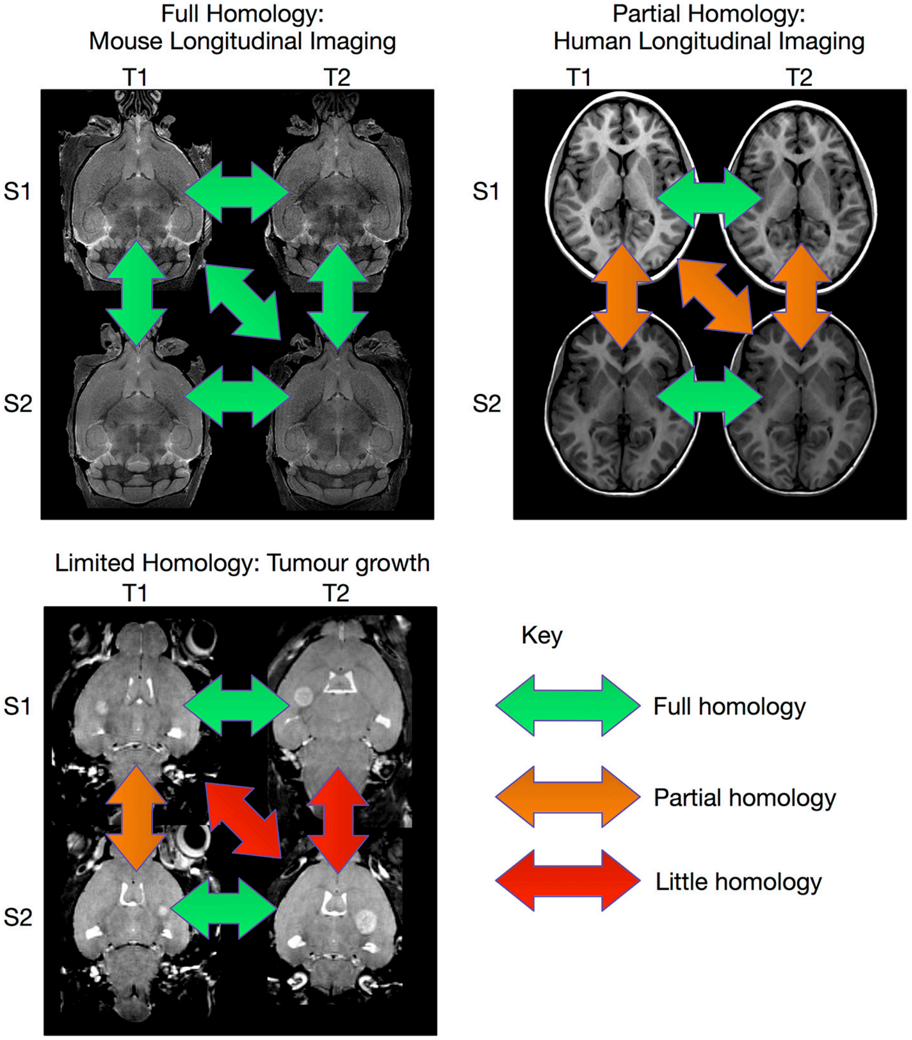


FIGURE 1 | Overview of registration scenarios. In the case of aligning cross-sectional adult mouse brains full homology exists between any pair of brains. Human longitudinal data, on the other hand, has full homology between scans of the same subject but more limited homology between different subjects. In the case of pathology, such as brain tumor growth, homology can only be found with a sufficiently sampled time-series, but is lost due to idiosyncratic tumor growth across subjects. (Tumor data courtesy Lisa Gazdzinski and Brian Nieman, The Hospital for Sick Children; see Gazdzinski and Nieman, 2014).

in preparation). The resulting transforms can be concatenated and used to calculate shape changes from a common coordinate space.

A hybrid of the two registration paradigms mentioned above can provide additional power to detect shape differences. Here longitudinally acquired scans from the same subject are aligned to each other, a per subject average image generated from those registrations, and these average images are then aligned across all subjects. This process allows for high fidelity registrations within subjects; after early brain development is complete and absent severe disease processes, homology across time within subjects is much higher than homology across subjects. This is particularly true with regards to ideosyncratic cortical folding patterns (Mangin et al., 2010). The second step of registering the per subject average images together then provides a common coordinate space so that the longitudinal data can be analyzed across the study population and, in so far as homology exists, shape differences across subjects computed.

Underlying the different types of image registration described above are many common features. The most obvious of these is the ability to align two brains to a common space, often in a multi-step procedure, and subsequently make use of the resulting output transform in a meaningful way. These transforms must be concatenated appropriately so that deformation fields can be calculated, regardless of the type of registration or common space. Moreover, as part of the registration process, brains must be resampled, derivatives calculated from the transforms, and segmented atlases brought into the common space of each study, to cite a few examples. Finally, in order for a registration to be successful, an underlying framework must be present to run each command in the appropriate order, keep track of dependencies (e.g., transforms must exist before they can be concatenated), output useful log files in case debugging is needed, and save the necessary files for statistical analysis in an organized fashion.

In this paper, we present Pydpiper: the computational framework we have developed to address these registration challenges. We wrote this toolkit with the following principles as paramount: (1) high-level coding should be as simple as possible for those with less coding experience (advanced users can still easily get "under-the-hood" to create new modules); (2) individual building blocks of code should be as modular as possible, easy to subclass, and geared toward a range of biologically relevant applications; (3) complete, runnable pipelines containing thousands of stages and addressing the registration scenarios described above should be available "out-of-the-box"; (4) at the end of any pipeline, there should be an option to calculate the derived volumes necessary for TBM based statistics, using a module that contains all of the required stages; (5) we should include a robust file handling class to keep track of naming schemes and file interactions across many modules in a single application. This class not only simplifies coding, but also allows seamless access to files created at any point in the pipeline. These principles influenced design choices all the way through our code hierarchy, including mechanisms of creating and combining pipelines as well as providing high level access to multiple image registration routines.

The rest of this paper is structured as follows. First, we will discuss existing neuroimaging software toolkits and describe and how Pydpiper fits into this space. Then, we will describe the underlying, application-independent pipelining framework that comprises Pydpiper; next, we will discuss the main levels of Pydpiper class structure, and how different classes may be pieced together to create new classes and applications; finally, we will describe in more detail the applications we have written to address four different registration challenges. To augment these sections, we include a worked example in section 5 that compares a registration pipeline written in Pydpiper with the corresponding code as it would be run manually on the command line. Finally, we conclude by highlighting the innovations Pydpiper brings to the existing space of pipelining frameworks used to solve neuroimaging problems.

2. MOTIVATION AND EXISTING SOLUTIONS

As described in the previous section, there are a number of commonalities that underlie seemingly disparate image registration strategies, all of which are frequently used in our group, and we wanted a toolkit to address all of them in a seamless way, focusing on the four core design principles listed above. Moreover, we found ourselves in a position that is common among many labs: frequently, a single executable and its related functions and libraries are coded to run one type of registration protocol and are not easily adaptable to other applications. In our case, we have used a highly successful pipeline environment, MICe-build-model (see https://wiki.mouseimaging.ca/display/MICePub/MICe-build-model), to do iterative, group-wise registration (Lerch et al., 2011), described both above and more fully in section 4.2. Unfortunately, using this tool to create any of the other types of pipelines was cumbersome, time-consuming and in many instances, was not fully-automated or required too many manual, intermediate steps. What's more, modification of code like this (whether written by us or others) can be prohibitively time consuming for neuroimaging students and post-docs who do not have an extensive computer science background. Finally, in our work on registration sensitivity (van Eede et al., 2013), we developed a set of optimized registration parameters for our iterative group-wise registration procedure. We wanted to adapt these and flexibly share them among different registration modules, but our existing tools did not allow for this.

There are a number of different software packages currently available for executing pipelines and building complex workflows, including VisTrails (Callahan et al., 2006), Taverna (Oinn et al., 2006), and Kepler (Ludäscher et al., 2006). Each of these packages provides both a comprehensive underlying framework and a graphical user interface (GUI) for constructing workflows; however, their aim is not to tackle problems specific to neuroimaging and they do not provide the extensive modules and support offered in other packages. This is in direct contrast to Pydpiper: here, the vast majority of our efforts were in constructing modules that are useful for solving neuroimaging registration challenges. The underlying framework, while a robust and necessary part of the toolkit, is not the main focus of Pydpiper.

Several frameworks have been written specifically to address the needs of the neuroimaging community. PSOM

(Bellec et al., 2012), written for Octave and Matlab, provides a pipelining overlay to direct scripting level programming, making complicated mathematical and statistical analyses easy to merge with pre-processing. The AIR (Woods et al. 1998a,b) package, written in C, provides source code and examples for running image registrations both within and across subjects and imaging modalities. LONI Pipeline (Dinov et al., 2010) is an extensive pipelining framework that, in addition to its robust underlying architecture, provides an elegant and user-friendly graphical user interface (GUI) for constructing pipelines. Another comprehensive and highly successful neuroimaging toolkit is Nipype (Gorgolewski et al. 2011), a Python-based, open-source software package. Both LONI and Nipype provide interfaces to many common neuorimaging tools such as SPM, FSL, and Freesurfer. These interfaces provide a powerful means for facilitating interactions between these packages. Comprehensive documentation and example scripts are also provided with both, so that users may construct and execute their own workflows.

Although the frameworks described above offer solutions to neuroimaging analysis problems, none of them addressed all of the design principles described in the previous section. For example, while both PSOM and AIR have functionality that overlaps with Pydpiper, PSOM is explicitly intended for developers and if one wants to utilize the source code directly, AIR requires a significant amount of user input and coding in order to execute complex, multi-step registrations[1]. This is in contrast to Pydpiper, which was designed to be accessible to researchers with little coding experience and runs four different types of pipelines upon installation. The GUIs offered by Taverna, VisTrails, Kepler, and LONI mitigate this issue to a degree, though users must still construct their workflows via “box and arrow” graph representations, and with the exception of LONI, were not written explicitly for neuroimaging applications. Even though each framework allows multistage pipelines to be combined into modules, this could still be cumbersome for pipelines with tens of thousands of stages. With Pydpiper, the existing building blocks are structured such that these dependencies are already built into the code, as will be discussed more in the following sections. In addition, because one of our goals was to create a toolkit that would enable non-programmers to write modules, we declined to write a GUI, which, in our experience, tends to dissuade people from exploring the code underneath.

In many ways, Nipype accomplishes much of what we intend to do with Pydpiper, is also written in Python and allows users to write their own code without needing to worry about the underlying architecture. It also provides additional functionality and interfacing that is not included in Pydpiper. As appropriate throughout this manuscript, we provide comparisons between Pydpiper and Nipype. We believe the two toolkits can provide complementary approaches for solving various image processing challenges. In the Discussion section, we outline both scenarios in which Pydpiper might be the preferred toolkit and scenarios where one would prefer Nipype.

Using the aforementioned design principles, Pydpiper was written with four specific applications in mind: (1) iterative, group-wise registration to create a study-specific average; (2) registration of adjacent time points in a chain-like fashion when all subjects cannot be registered together; (3) two-level registration for longitudinal studies where both subject-specific and study-specific averages are created; and (4) an automated multi-atlas label generation procedure. To assist in reusability, Pydpiper provides class types to manage distinct aspects of pipeline creation: “atoms” wrap distinct operations (e.g., registering two images), “modules” link together atoms into reusable processing subunits, and “applications” provide a command-line interface allowing users to drive a particular pipeline. In addition, we created a comprehensive file handling framework to simplify future code development and usage of these atoms and modules. All of this was done with the overarching goal that atoms and modules could be easily combined to create entirely new types of registration pipelines. Moreover, Pydpiper is specifically designed to take advantage of grid computing environments and automatically calculates stage dependencies, decreasing the time necessary for both coding and execution.

In addition to the aforementioned design considerations, we wanted Pydpiper to be a tool that is freely available to the community, with low barriers for adaptation and usage by others. This not only has the effect of continually improving upon Pydpiper, but also increases both transparency and reproducibility of results obtained by using it (Ince et al., 2012). It is distributed under the Modified BSD license, which allows free copying, modification and distribution of the code and is freely available on github (https://github.com/mfriedel/pydpiper). This distributed version control system (git) allows for the tracking of all changes, a complete history of the source code, and the ability to flag issues and discuss them with other developers. As a companion to this paper, a public wiki is also available and contains more detailed information about development, usage and applications. (https://wiki.mouseimaging.ca/display/MICePub/Pydpiper) A virtual machine for code testing and example workflow diagrams are included as well. Additionally, Pydpiper is written in Python and uses the Pyro (https://pypi.python.org/pypi/Pyro4) and NetworkX (http://networkx.github.io/) libraries, all of which are freely available, straightforward to install and enjoy broad support and usage. Pydpiper has been developed for the Linux operating system, the most popular platform currently in use by the neuroimaging community (Hanke and Halchenko, 2011). Finally, we wanted to create a toolkit that could be easily used without extensive programming knowledge. While we welcome and encourage contributions to Pydpiper from expert developers, we structured the classes and example applications such that someone with only a basic knowledge of Linux, Python and Object Oriented Programming could create a pipline specific to their needs.

3. DESIGN AND IMPLEMENTATION

3.1. GENERAL PIPELINE AND APPLICATION STRUCTURE

The core Pydpiper framework that serves as the base for all applications was designed to be as modular and reusable as possible. It is also completely independent of the application

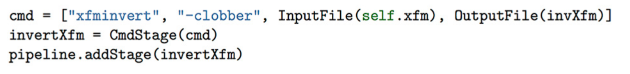
[1] We note there that AIR is a module that can be used within LONI. In this instance, we are talking about compiling and using the source.

being executed. Although we have written this toolkit with an image registration focus, the framework that manages pipeline construction and execution could be used for any type of software engineering paradigm that follows a similar design pattern. This framework is encapsulated in five core classes: *PipelineStage, CmdStage, Pipeline, AbstractApplication*, and *pipelineExecutor*. Taken together, they act in concert to construct pipelines with one or more stages, connect them through a series of interdependencies, execute each stage in the appropriate order via thread pool and encapsulate each pipeline into a larger application that is executed on the command line.

PipelineStage is the primary base class upon which all additional executable classes are built. It was designed to contain all of the underlying framework necessary to successfully integrate a single stage into a larger pipeline. This framework includes identifying inputs and outputs, creating and writing to a log file, and keeping track of both stage status (e.g., running, finished, failed) and the amount of memory and processors required for execution. *PipelineStage* also contains the functions that get and set the amount of memory and processors needed for a particular stage as well as those needed for setting the status of a stage (e.g., running, finished, or failed).

The command stage (*CmdStage*) class inherits directly from *PipelineStage*. The primary difference between *CmdStage* and *PipelineStage* is that pipeline stages can run arbitrary pieces of Python code, while command stages are designed to execute individual command line programs. Although our current applications rely heavily on the *CmdStage* functionality, we explicitly wrote *PipelineStage* as the base class, so that Pydpiper users can include pieces of code that don't necessarily require command line execution.

The arguments necessary for running a command, as well as the command itself, are passed to *CmdStage* as an array, appropriately parsed. The command is then executed at the appropriate time using the Python function `call`. Any command line executable that is called as part of a larger pipeline must be an instance of *CmdStage* and each command stage can run only a single command line executable. Although many command stages are subclassed, as will be described further in section 3.2, they can also be constructed on the fly. If there is a command-line executable that is used only once (and therefore does not warrant its own subclass of *CmdStage*) an array of input and output files can easily be converted to a command stage as shown in **Figure 2**.



```
cmd = ["xfminvert", "-clobber", InputFile(self.xfm), OutputFile(invXfm)]
invertXfm = CmdStage(cmd)
pipeline.addStage(invertXfm)
```

FIGURE 2 | Example of how to construct an executable Pydpiper stage using the `CmdStage` class. The example command used to construct this stage is `xfminvert`, which takes a transform between two subjects and inverts it. (`xfminvert` is part of the MINC toolkit, described more fully in section 3.2. A more complete usage example is also provided in section 5). After instantiating the class, it is added to the pipeline via the `addStage` function. Note that `InputFile` and `OutputFile` are themselves classes, designed to indicate to CmdStage the required inputs and outputs for stage interdependencies.

A pipeline (*Pipeline*) is composed of any number of pipeline and/or command stages, and as such, the *Pipeline* class tracks dependencies between stages and keeps a queue of runnable stages and stage state. One of the most critical features of this class is that it infers stage interdependencies based on stage inputs and outputs. That is, if one or more output files from stage A are required for stages B and C, *Pipeline* keeps track of this dependency, and does not add stages B and C to its queue of runnable stages until stage A is complete. Conversely, stages may be executed in any order once all of their dependencies have been satisfied. To capture stage connectivity, the NetworkX library (http://networkx.lanl.gov/) is used to implement Pydpiper pipelines as a directed graph. In addition to the *addStage* command shown as part of **Figure 2**, *Pipeline* also provides a function called *addPipeline* allowing pipelines to be combined, increasing the ease with which modular code can be written. When stages are added to a pipeline, they are skipped if they already exist. This not only shortens run times, but makes Pydpiper code itself easier to write and read. An example of this type of coding can be found in section 3.2.

In addition to maintaining a queue of runnable stages, *Pipeline* tracks the state of each of its stages (running, finished, or failed). The Pipeline class also uses the Python pickling mechanism, a standard means of object serialization, to save essential pipeline features after each completed stage. This allows an unfinished pipeline to easily be restarted from pickled backup files. The following data is pickled: the directed graph describing stage interdependencies; an array of pipeline stages; the current stage counter; a hash uniquely identifying each stage; a hash of output files for each stage; and an array containing the statuses of each stage. To restart a pipeline, one would simply specify `--restart` as a command line option when launching pipeline executors, as described below. The `--restart` option will then load the pickled data into the appropriate variables before starting the pipeline. The graph heads and edges can be quickly reconstructed by iterating through the saved and reloaded directed graph, and all stages with "finished" status are not re-run.

Because of the directed graph architecture of pipelines like this, many stages can be run in parallel, provided their predecessor stages have completed successfully. To run these stages most efficiently, we created the *pipelineExecutor* class. Pipeline executors are managed as a thread pool, with each thread executing individual stages from the pipeline's runnable stages queue. These executors effectively act as clients to the pipeline, which functions as a server. The number of executors required, threads per executor and memory necessary for each process are specified on the command line. Executors can be launched independently, as a stand alone command, or they can be launched as part of an application itself. The values chosen with respect to memory and processors will vary both with an application and available computational resources. Each executor is then initialized as a client of the pipeline server. This client/server architecture is implemented using the Python Remote Objects (PYRO) library (https://pypi.python.org/pypi/Pyro4), and support is included for running on clusters with both the pbs and sge queueing systems. By specifying either `--queue=pbs` or `--queue=sge`, Pydpiper will create a script with the appropriate syntax and automatically submit it to the requested queue. For example, by including `--queue=pbs --ppn=8 --num-executors =1 --proc=8 --time=18:00:00`, Pydpiper will create and submit a pbs script requesting a single node with 8 processors (via `--ppn`). Once running, this script will launch a single executor with eight threads that will run for a maximum of 18 h.

One of the most salient features of pipeline executors is how they interact with the pipeline. Each executor can consist of one or more threads. In turn, each thread will poll the server to get the next available stage from the pipeline's queue of runnable stages. If enough memory and processors are available to run that stage, the thread will execute the stage. Otherwise, it will sleep for a specified interval before re-polling the server. Once a stage has finished running (or failed to complete), the thread will release the memory and processors used and poll the server again for the next available stage to run. This happens repeatedly by all threads until all stages in the pipeline have finished. Alternatively, if there are failed stages, the pipeline will shut itself down once no more stages can be run. (In this instance, debugging will be necessary before restarting the pipeline). In addition, if an insufficient number of executors were launched, additional executors may be launched at any time via the command line. This may be done whether running locally, or if using an sge or pbs supported cluster.

To tie together command stages, pipelines and pipeline executors into a single runnable program, we created the abstract application (*AbstractApplication*) class. This is the base class for all applications written within the Pydpiper framework. Each class that inherits from *AbstractApplication* will itself be a command line executable that, when launched with the appropriate arguments, will run an entire pipeline from start to finish. This class sets up command line options that are required for all subclasses, initializes the pipeline (or restarts it from backup files) and sets up a logger. It also launches the pipeline daemon, which is where the pipeline is initialized as a server. If the appropriate command line options are specified, subclasses of *AbstractApplication* will launch executors, so that they may begin running immediately. When writing a new application that inherits from *AbstractApplication*, one only needs to extend a few functions without having to worry about the underlying framework. These functions are shown in **Figure 6**. A more complete example of a Pydpiper application that inherits from *AbstractApplication* is included in the section 5.

3.2. CLASS HIERARCHY AND FILE HANDLING

As noted in the Introduction, Pydpiper supports three main "levels" of classes that are built on top of the core Pydpiper framework described above: atoms, modules and applications. In addition, there is a file handling framework to help simplify their usage. All of the initial classes we developed extend the Pydpiper framework to support files and pipelines that use the Medical Imaging NetCDF (MINC) file format. MINC is a comprehensive medical imaging data format and an associated set of tools and libraries. It was initially developed at the Montreal Neurological Institute (MNI) and is freely available online. (http://www.bic.mni.mcgill.ca/ServicesSoftware/MINC, http://en.wikibooks.org/wiki/MINC). In addition, we make use of pyminc, a Python interface to the MINC2 library (https://github.com/mcvaneede/pyminc). We expect that as development continues (by both us and other members of the community) other file formats will be supported as well.

Pydpiper atoms inherit directly from *CmdStage* and act as wrappers around frequently used MINC tools. Each atom has at least one required argument, an input MINC file, which may be passed as a string or a file handler. Additionally, most atoms require a second argument, a target MINC file, which must be passed in the same format (e.g., string or file handler) as the input MINC file. As is noted in the Introduction, image registration determines the transformation mapping one image (source) into the space of another (target), and Pydpiper's atomic structure reflects this. All atoms have multiple optional arguments which are either specified directly or make use of the `**kwargs` functionality built directly into Python. The choice of optional arguments, and their defaults, were selected based on the most common ways in which we use the MINC tools. An example of minc atom usage is shown in **Figure 3**. This figure depicts two different ways to call the *mincANTS* atom. This atom calls the command-line program of the same name, the MINC-based implementation of the Advanced Normalization Tools (ANTs) (Avants et al., 2008), a diffeomorphic image registration software package. Whether only two file handlers are specified or the entire list of optional arguments is included, the atom will handle putting together the command to be executed and, because it inherits from *CmdStage*, all of the attributes necesssary to seamlessly integrate it into an existing pipeline are present.

As discussed above, a critical component of running any type of pipeline is keeping track of stage dependencies, inputs and outputs. As is typical of the neuroimaging pipelines that formed the motivation for Pydpiper, each input image in a pipeline is related to others via a series of registrations, transforms and resampling. In addition to stage interdependencies, one also needs to keep track of, for example, the most recent transform between any two images. Or, if a file has been resampled, it may be necessary at a later point to access the original version of the file. Keeping track of these files can be cumbersome, particularly for novice developers, and doing so without resorting to unnecessarily repetitive code can be a challenge. To address this challenge, we have created the *RegistrationPipeFH* class, and its parent class, *RegistrationFHBase*. Each input scan used in a pipeline (typically read in as a command line argument) can be initialized as a file handler (i.e., as an instance of the *RegistrationPipeFH* class). A more complete discussion of how file handlers are instantiated is included in section 5. Although this is not a requirement for using Pydpiper, by using file handlers, all future use of a given input is dramatically simplified. In addition, this class makes it easier to identify the appropriate inputs and outputs to individual stages when constructing new command stages and atoms.

One of the key features of file handlers is the way that they allow access to the state of an image at any stage in the pipeline, and various transforms or resampled files can be retrieved at any time for later use. As a more specific example this, consider the *minctracc* atom, which registers two files based on a specified set of parameters. This atom serves as a wrapper for minctracc, the implementation of the ANIMAL non-linear registration

```
ma = mincANTS(inputFH, targetFH)
pipeline.addStage(ma)
```

```
ma = mincANTS(inputFH,
              targetFH,
              defaultDir="tmp",
              blur=[-1, 0.056],
              gradient=[False, True],
              similarity_metric=["CC", "CC"],
              weight=[1,1],
              iterations="100x100x100x150",
              radius_or_histo=[3,3],
              transformation_model="SyN[0.05]",
              regularization="Gauss[3,0]",
              useMask=True)
pipeline.addStage(ma)
```

FIGURE 3 | Simplified call of the *mincANTS* atom (left) and a call that includes all arguments (right). The call on the left requires only an input and target file handler, and uses default arguments as *mincANTS* parameters. On the right is a *mincANTS* call that includes specific arguments as parameters, overriding the defaults. These arguments correspond to various command line options required by *mincANTS* and they are discussed in more detail in section 5. We also refer the reader to Avants et al. (2008) and references therein for a complete discussion.

method (Collins et al., 1994, 1995). Although an extensive number of optional minctracc arguments exist, the only requirements for this atom are an input and target. If this input and target are file handlers, minctracc will retrieve the appropriately blurred version of this file (created previously and saved in a dictionary by the file handling class), and set the output transform as the subsequent last transform between input and target, so it can easily be retrieved later if desired. Moreover, if several minctracc calls are made in succession on the same two files, the file handling class will keep track of all previous transforms while still "knowing" which one was the most recent. This results in increasingly simple function calls, particularly within more complex modules. Additionally, any of these transforms can be retrieved at any point in the registration process. An example of this is shown in **Figure 4**.

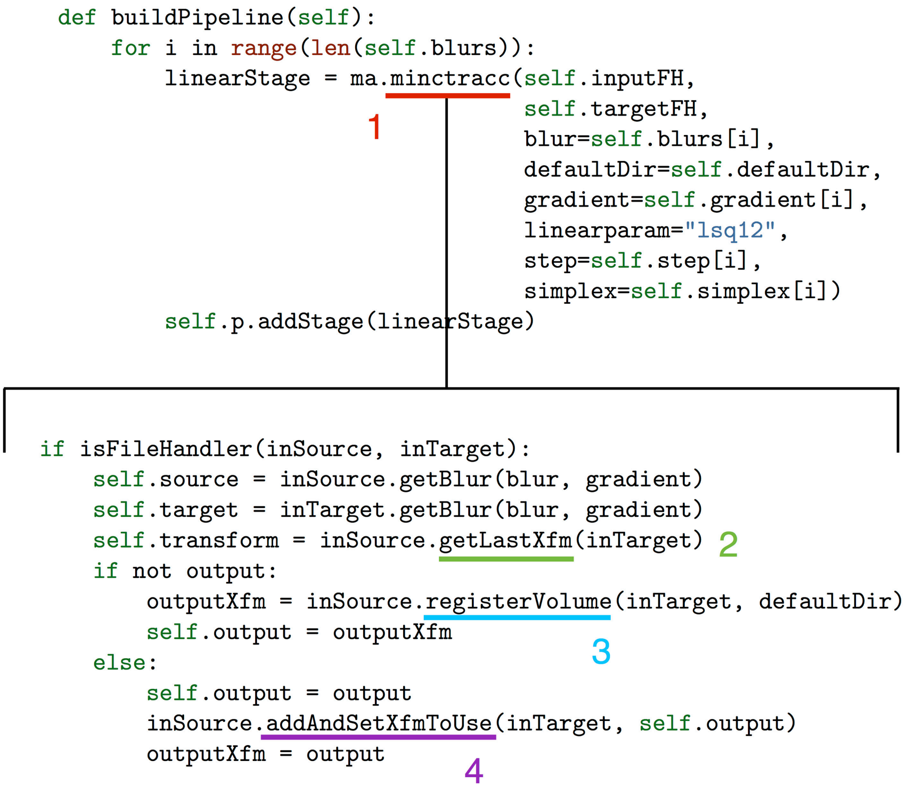
Modules are perhaps the most flexible and essential component of the Pydpiper toolkit. A module can be composed of a multiple atoms and command stages or a combination of atoms and other modules. Existing modules were designed such that they can be easily pieced together and used in multiple types of pipelines, even for applications that at first glance seem to have quite different architecture. A good example of a Pydpiper module is the *HierarchicalMinctracc* class pictured in **Figure 5**. This class calls both atoms and other modules and can be easily subclassed or called as is. Including *HierarchicalMinctracc* in a larger pipeline is as simple as instantiating this class as part of a larger module or application (`hm = Hierarchical Minctracc(inputFH, targetFH)`) and adding it to the existing pipeline (`p.addPipeline(hm.p)`). Additional arguments (as shown in the `__init__` in **Figure 5**) can be included when the class is called, but are not required.

We noted in section 3.1 that coding with Pydpiper can be done in a non-linear fashion, such that stages in the pipeline are skipped if they already exist. One example of this is depicted in **Figure 5**. On lines 52–53 of the code, we blur the images associated with `inputFH` and `targetFH`. This is done once for each of the blurs specified in the non-linear protocol (`self.nlin_protocol`), itself defined in the `__init__` function. These blurred images are then registered together, by the `minctracc` call on line 58. (The rationale for blurring is described in more detail in the following section). It is often the case, however, that `HierarchicalMinctracc` is called in a loop, once for many different input images (each with their own file handler, `inputFH`) all registered toward the same target (`targetFH`). Because the same set of blurs is often used, this means that line 53 will construct the exact same pipeline stage multiple times. However, within `addStage`, there is a check to see if the pipeline already contains an instance of this stage. If it does, the stage is not added again to the pipeline. This results in code that is easy to read (it is conceptually simple to understand why one would want to execute the same command on both an input and target) and write (the programmer does not need to keep track of whether or not the target file has already been blurred in a previous instantiation of `HierarchicalMinctracc`).

Applications build on both atoms and modules to provide a complete implementation of a single pipeline. The essential feature of an application is that it is a command line executable that inherits from the *AbstractApplication* class described in section 3.1. In theory, an application can be as simple as a single pipeline stage, or one with thousands of stages that are constructed through multiple atoms and modules. Although the complete pipeline for a given application can be extremely complex, at its highest level the application code was designed to be quite simple. This is shown in **Figure 6**. A more detailed description of each of Pydpiper's current main applications is included in the following section.

4. EXAMPLE APPLICATIONS

In section 1, we briefly introduced the scientific rationale for the applications that motivated the development of Pydpiper. As is noted there, different experimental designs require different



```
def buildPipeline(self):
    for i in range(len(self.blurs)):
        linearStage = ma.minctracc(self.inputFH,
                                   self.targetFH,
                                   blur=self.blurs[i],
                                   defaultDir=self.defaultDir,
                                   gradient=self.gradient[i],
                                   linearparam="lsq12",
                                   step=self.step[i],
                                   simplex=self.simplex[i])
        self.p.addStage(linearStage)
```

```
if isFileHandler(inSource, inTarget):
    self.source = inSource.getBlur(blur, gradient)
    self.target = inTarget.getBlur(blur, gradient)
    self.transform = inSource.getLastXfm(inTarget)
    if not output:
        outputXfm = inSource.registerVolume(inTarget, defaultDir)
        self.output = outputXfm
    else:
        self.output = output
        inSource.addAndSetXfmToUse(inTarget, self.output)
        outputXfm = output
```

FIGURE 4 | The buildPipeline function that is part of one of the Pydpiper modules (top) and a portion of the highlighted minctracc class (bottom). The minctracc class (1), called multiple times in the for loop, is expanded to show details about how the file handling classes operate. Each time minctracc is called, getLastXfm (2) finds the last transform between input and target and uses it as the input transform for the current function call. If no previous transform exists, an appropriate default is set based on the specified registration parameters. If an output transform is not specified as an argument when minctracc is called (as in this example), registerVolume (3) creates the output file name based on a set of defaults that includes the input and target names and whether or not a previous transform exists between these files. If an output transform is specified, addAndSetXfmToUse (4) adds this transform to the dictionary of transforms between input and target. If the blurs, gradient, step and simplex are not specified when minctracc is called, defaults will be used.

registration paradigms. This is particularly true when considering whether and how a common space for all subjects should be created. Nevertheless, commonalities that underlie seemingly disparate registration strategies are largely what shaped the design and development of Pydpiper. In this section, we will describe these common features in more detail and then discuss how they are combined in various ways to address specific image registration challenges.

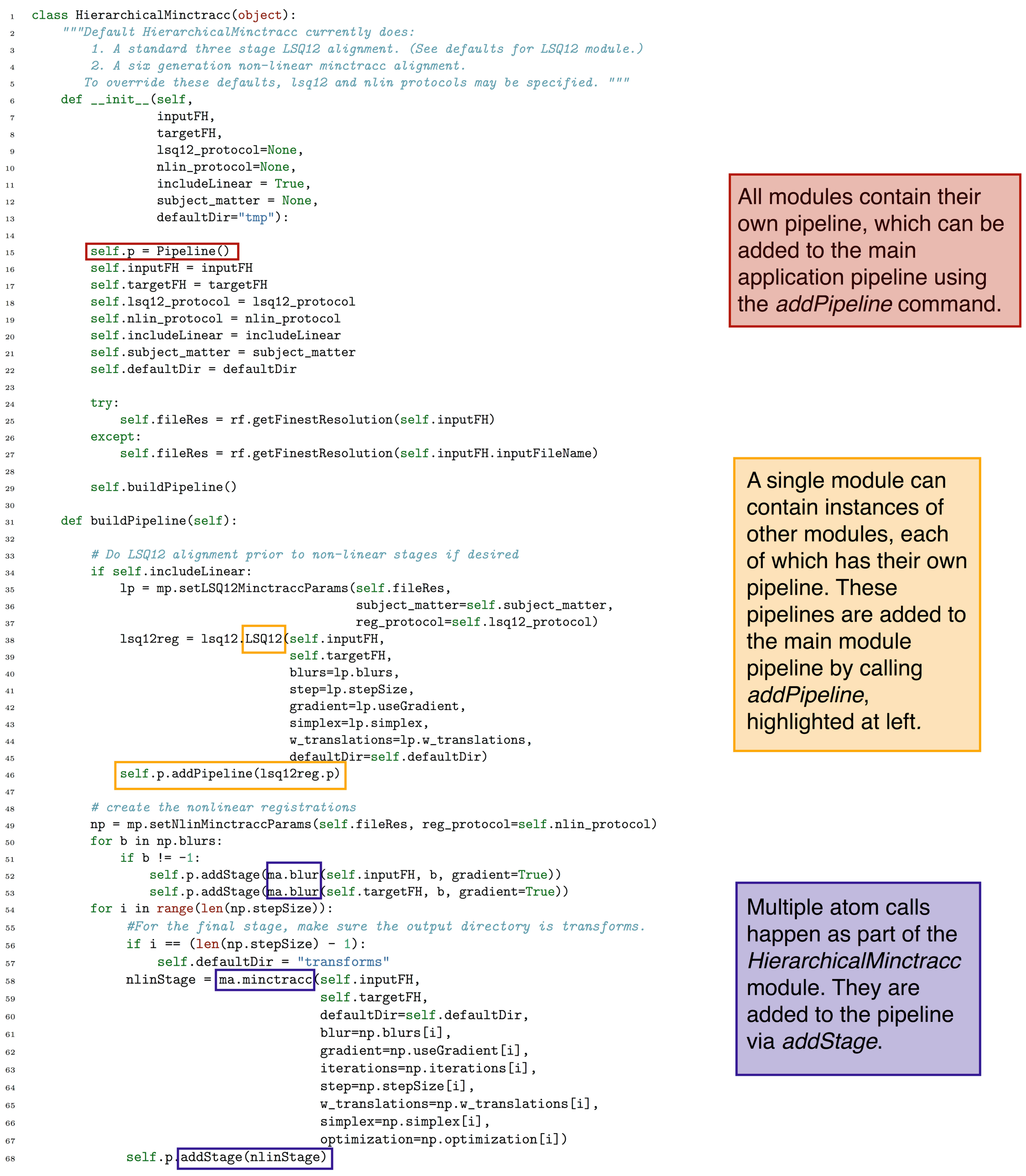
4.1. ESSENTIAL REGISTRATION MODULES

4.1.1. LSQ6

Each input image in a given study is scanned in a slightly different coordinate system, and prior to more precise alignment, it is beneficial if all scans are in the same coordinate system. This happens by applying translations and rotations to each image to align them toward a common target. This common target can be one of the input images, or a specified initial model that is in the desired coordinate system. Because this type of alignment involves six degrees of freedom (three translations and three rotations), we refer to it as LSQ6. For each brain, LSQ6 involves the following steps: (1) blur each input image with a specified Gaussian smoothing kernel (necessary so as not to overly weight singularities or extreme inhomogeneties in an image) (2) align, with a specified registration algorithm, each of the blurred images (3) repeat steps 1 and 2, if desired, for a series of different blurs and (4) resample each input brain with the transform generated from stage 3. The Pydpiper LSQ6 module wraps all of these stages (each of which is its own minc atom) inside a single class. This class takes an array of file handlers (one for each input image in the study) and applies this alignment to each of them.

4.1.2. LSQ12

Whether or not an LSQ6 alignment is required, the next step (or first step) in registering images is often to create an affine alignment between a source and target. This typically involves aligning the source and target via a series of translations, rotations, scales and shears. Because each of these deformations contributes three degrees of freedom, we call this stage of registration LSQ12. Depending on the type of registration pipeline, LSQ12 can be used in different ways. If all subjects in a study are being registered together, it can be beneficial to do an LSQ12



```
class HierarchicalMinctracc(object):
    """Default HierarchicalMinctracc currently does:
        1. A standard three stage LSQ12 alignment. (See defaults for LSQ12 module.)
        2. A six generation non-linear minctracc alignment.
       To override these defaults, lsq12 and nlin protocols may be specified. """
    def __init__(self,
                 inputFH,
                 targetFH,
                 lsq12_protocol=None,
                 nlin_protocol=None,
                 includeLinear = True,
                 subject_matter = None,
                 defaultDir="tmp"):

        self.p = Pipeline()
        self.inputFH = inputFH
        self.targetFH = targetFH
        self.lsq12_protocol = lsq12_protocol
        self.nlin_protocol = nlin_protocol
        self.includeLinear = includeLinear
        self.subject_matter = subject_matter
        self.defaultDir = defaultDir

        try:
            self.fileRes = rf.getFinestResolution(self.inputFH)
        except:
            self.fileRes = rf.getFinestResolution(self.inputFH.inputFileName)

        self.buildPipeline()

    def buildPipeline(self):

        # Do LSQ12 alignment prior to non-linear stages if desired
        if self.includeLinear:
            lp = mp.setLSQ12MinctraccParams(self.fileRes,
                                            subject_matter=self.subject_matter,
                                            reg_protocol=self.lsq12_protocol)
            lsq12reg = lsq12.LSQ12(self.inputFH,
                                   self.targetFH,
                                   blurs=lp.blurs,
                                   step=lp.stepSize,
                                   gradient=lp.useGradient,
                                   simplex=lp.simplex,
                                   w_translations=lp.w_translations,
                                   defaultDir=self.defaultDir)
            self.p.addPipeline(lsq12reg.p)

        # create the nonlinear registrations
        np = mp.setNlinMinctraccParams(self.fileRes, reg_protocol=self.nlin_protocol)
        for b in np.blurs:
            if b != -1:
                self.p.addStage(ma.blur(self.inputFH, b, gradient=True))
                self.p.addStage(ma.blur(self.targetFH, b, gradient=True))
        for i in range(len(np.stepSize)):
             #For the final stage, make sure the output directory is transforms.
             if i == (len(np.stepSize) - 1):
                 self.defaultDir = "transforms"
             nlinStage = ma.minctracc(self.inputFH,
                                      self.targetFH,
                                      defaultDir=self.defaultDir,
                                      blur=np.blurs[i],
                                      gradient=np.useGradient[i],
                                      iterations=np.iterations[i],
                                      step=np.stepSize[i],
                                      w_translations=np.w_translations[i],
                                      simplex=np.simplex[i],
                                      optimization=np.optimization[i])
             self.p.addStage(nlinStage)
```

All modules contain their own pipeline, which can be added to the main application pipeline using the *addPipeline* command.

A single module can contain instances of other modules, each of which has their own pipeline. These pipelines are added to the main module pipeline by calling *addPipeline*, highlighted at left.

Multiple atom calls happen as part of the *HierarchicalMinctracc* module. They are added to the pipeline via *addStage*.

FIGURE 5 | Code snapshot of the `HierarchicalMinctracc` class. In this class, there are calls to both atoms (e.g., `blur` and `minctracc`) and modules (`LSQ12`). Note that `minctracc` is called iteratively, as is shown in **Figure 4**, but is using a different subset of arguments.

registration between all pairs of subjects in the study (Kovačević et al., 2005) immediately following the LSQ6 alignment. This proceeds similarly to LSQ6: a single LSQ12 call between two brains involves a series of blurs and alignments, with a final resampling of each subject at the end. The goal of this procedure is the creation of an average of all subjects in LSQ12 space. In other types of pipelines, a full pairwise LSQ12 registration is not appropriate due to insufficient homology among subjects, but an LSQ12 alignment between specific sets of subject/template pairs can improve registration accuracy. The Pydpiper LSQ12 module handles both of these instances from a common class.

4.1.3. NLIN

In many ways, the most critical step of image registration is non-linear alignment. This is typically the final stage of image registration, and involves non-uniform deformation of a source image to a target, optimized via a particular metric. In contrast to the LSQ6 and LSQ12 modules previously described, in which all voxels

```
class RegistrationChain(AbstractApplication):
    def setup_options(self):
        # Options setup here

    def setup_appName(self):
        appName = "Registration-chain"
        return appName

    def run(self):
        # Code here to setup and run pipeline


if __name__ == "__main__":

    application = RegistrationChain()
    application.start()
```

FIGURE 6 | Example of Pydpiper application code. Along with the required import statements (omitted from this figure for brevity), the `.py` file necessary to create an executable for a given application is extremely simple. This example is for the RegistrationChain application described in section 4.3 and is representative of how to construct an application that inherits from `AbstractApplication`. There are three functions included in `RegistrationChain`: `setup_options`, `setup_appName`, and `run`. `run` is the function that calls a unique combination of Pydpiper atoms and modules to construct the appropriate pipeline and in spite of the complexity inherent in this type of registration, this function is less than 100 lines of code. At the end of the file the `if __name__ = "__main__"` clause is required so that this code can be executed directly from the command line. In section 5, we show a complete example, albeit for a different application, of these functions.

are deformed in a uniform, global way, non-linear registration induces non-uniform deformations. When all scans in a study can be registered together, non-linear registration may happen iteratively, toward an evolving target. After each subject is registered to an initial target (for instance, the LSQ12 average), all subjects are resampled, a new average is created, and alignment proceeds to this new average. Alternatively, a single subject/template pair could be non-linearly aligned with either a single or multi-stage call, but without iterating toward an evolving target. Examples of this include the registration chain paradigm (described in section 4.3) and multiple automated template generation (section 4.5). One of the design goals of Pydpiper was to create a series of non-linear modules that handle either of these registration scenarios in a straightforward way. Moreover, there are multiple different types of non-linear registration metrics that are available (Klein et al., 2009), including Advanced Normalization Tools (ANTs) (Avants et al., 2008) and Automatic non-linear Image Matching and Anatomical Labeling (ANIMAL) (Collins et al., 1994, 1995), the two algorithms we have utilized in Pydpiper. Although they differ significantly "under the hood," (elastic vs. diffeomorphic optimization, completely different command line options) one of our goals was to implement them such that their usage at a high level is nearly identical. The ANTs toolkit itself provides a number of helpful bash scripts for various types of image alignments, including the type of iterative model building described in section 4.2. However, by incorporating this same paradigm directly into the Pydpiper framework, we have greater flexiblity to use it in conjunction with other Pydpiper modules. In addition, our file handling framework makes it easier to access files created throughout the entire registration process, something that would require additional scripting if using the ANTs toolkit as a stand-alone package.

4.1.4. Pre-processing

In addition to the LSQ6 and LSQ12 modules, there are several pre-processing steps that often need to be included before proceeding with non-linear registration. The most important of these is applying a non-uniformity correction to each image to account for smooth intensity variations that are often present in MR imaging of homogenous tissue (Sled et al., 1998). Another pre-processing stage is intensity normalization, which addresses interslice intensity variations (Zijdenbos et al., 1995). Although each of these steps are most sensibly applied prior to non-linear registration, our goal was to code them such that they could be called at any stage of any type of pipeline. In addition to both of these steps, another step that may be critical to a successful registration is masking. MRI scanning, particularly when done *ex-vivo*, can result in images where a non-negligible amount of tissue is present around the outside of the brain. In order to speed up the registration process and increase its accuracy, a region of interest is defined that encompases the entire brain, and image alignment only occurs within this region. Defining and keeping track of masks and using them when appropriate was also a key feature included in our design and development of Pydpiper, particularly with respect to the file handling class described previously.

4.1.5. Statistics

Finally, the end-goal of performing statistical analysis based on the results of a registration, regardless of type, factored heavily into the design of Pydpiper. For many types of registrations, all statistical analysis must be done from a common space, but how this common space is constructed varies with the type of pipeline. Once a common space has been identified, the full transform from this common space back to each individual subject is used to calculate a deformation field. After smoothing and taking the Jacobian determinant of this deformation field (a measure of the volume expansion or contraction at each voxel) we can use DBM to calculate neuroanatomical differences due to genotype, gender, environmental factors, etc. In particular, the statistics module of Pydpiper was designed with two paradigms in mind: the first was that once the appropriate transform was identified, the calculation of the associated deformation field and Jacobian determinants would proceed as uniformly as possible; the second was that the transform concatenation often necessary to get the appropriate average-to-subject transform would happen in a modular way, independent of determinant calculation, to increase code reusability. This was motivated in part by differences between iterative group-wise registration (section 4.2) and the registration chain (section 4.3). In the latter, deformation fields can be calculated both from a space common to all subjects, or between individual subject pairs, and we wanted code that would handle both in a seamless fashion, particularly at the highest levels.

4.2. ITERATIVE GROUP-WISE REGISTRATION

Our previous implementation of iterative group-wise registration is described in more detail in Lerch et al. (2011). In Pydpiper, we utilized the same underlying logic and theoretical framework for this application, but implemented it in a much more streamlined and extensible fashion. Briefly, this iterative, group-wise registration proceeds as follows: we first bring all subjects into a common space using the LSQ6 module. Then, following non-uniformity correction and intensity normalization, we perform a pairwise registration of all subjects in the study using the LSQ12 module. This creates the best possible linear model for this data set. Using the LSQ12 average as a starting template, we then locally deform each scan toward this template, using either an elastic (minctracc, Collins et al., 1994, 1995) or diffeomorphic (mincANTS, Avants et al., 2008) registration algorithm. After this initial alignment, another average is created, and this is used as a template for subsequent non-linear generations. This entire multi-generation procedure is encapsulated in the non-linear (NLIN) registration module. Once a final non-linear average is created, the appropriate transforms are concatenated and used to create deformation fields from this template to each individual subject. These deformation fields are subsequently used in DBM. A schematic of this registration process is depicted in **Figure 7**. A corresponding code diagram is shown in **Figure 8** and the annotated code itself is provided in **Figure 9**.

One notable feature of our implementation of iterative group-wise registration is that, at a high level, the code is deliberately sparse. The goal of this design was to make each stage (e.g., LSQ6, LSQ12, NLIN, statistics calculations) an independent entity, to aid in both readability and provide a more direct correspondence between the theoretical framework and the code itself. As an example of the size of one of these pipelines, consider an image registration with 10 mutants and 10 wild type mice, the minimum number we typically use for a two group comparison. This pipeline would have a total of 2169 pipeline stages encapsulated into four modules: LSQ6 (including intensity normalization and pre-processing), LSQ12, NLIN and Statistics. For larger studies and the alternate strategies described below, (particularly MAGeT), pipelines can often consist of tens of thousands of stages; however, because of the modular nature of the code, applications remain uncluttered and easy to read.

The modular nature of Pydpiper applications also makes it easier to assess where changes to the pipeline should occur. For example, one might want to proceed directly to non-linear registration after having performed the LSQ6 stage–this could be done quite simply by removing only a few lines of code in the existing application. In addition, each of these modules has a default set of registration parameters that are based on the detected input file resolution. Alternate parameters may be deliniated in a .csv file that is specified on the command line when the application is launched. This makes it simple to flexibly adjust parameters as needed while avoiding hard coded values that are only appropriate for a handful of cases. Another advantage of this modular code is that it is simple to implement alternate registration strategies. For example, the non-linear modules (NLIN) for both minctracc and mincANTS registrations inherit from a common base, which could easily be further subclassed to create an alternate non-linear registration strategy.

4.3. REGISTRATION CHAIN

There are numerous scenarios where the iterative group-wise registration paradigm described in the preceeding section is inappropriate, and alternative registration and analysis strategies must be employed. This is particularly true in the case of specific types

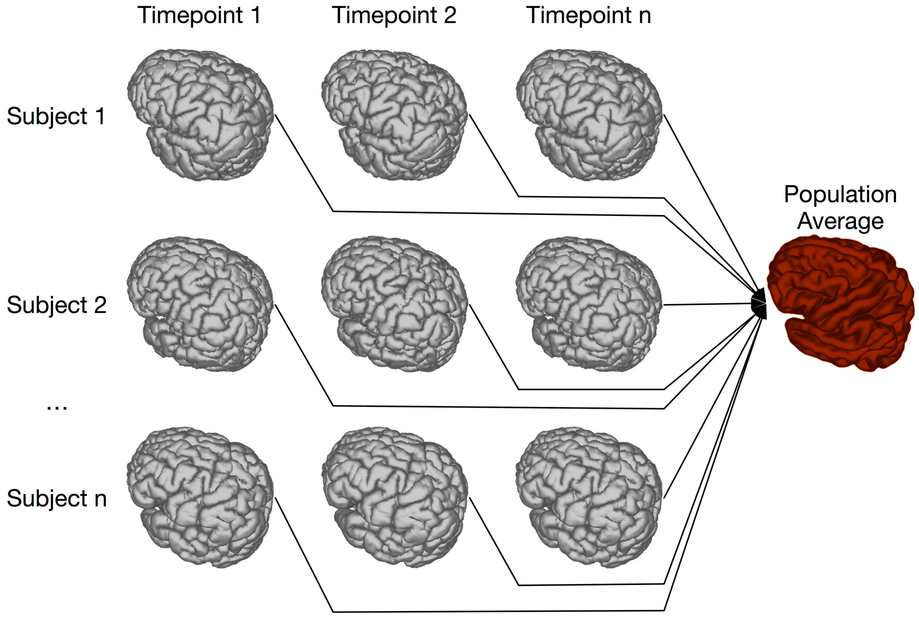


FIGURE 7 | Schematic of iterative group-wise registration. This schematic depicts a registration scenario where all subjects, each scanned at one or multiple time points, can be registered to a consensus average.

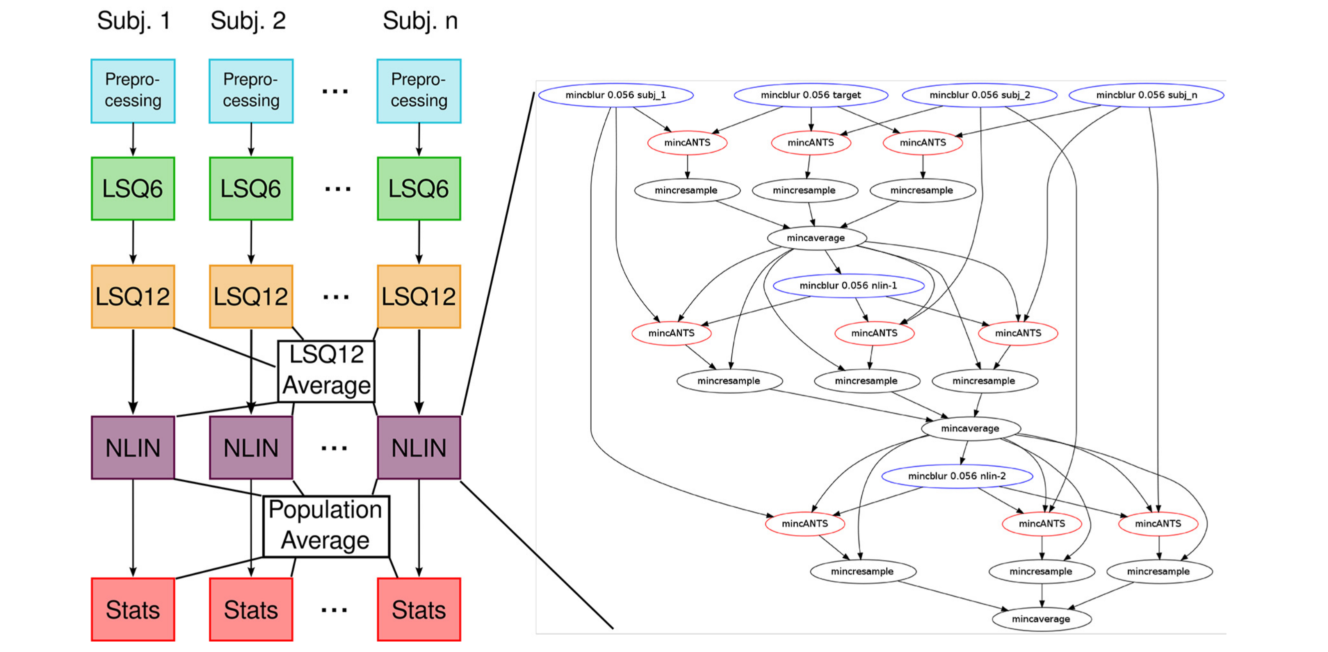


FIGURE 8 | Code diagram to complement Figures 7, 9. This figure illustrates the modular nature of how Pydpiper executes each of the stages in this pipeline. Diagramatically, each of the code blocks highlighted in **Figure 9** is indicated here as a single unit. One of the non-linear stages is expanded to show the complexity of the pipeline that underlies it.

of longitudinal studies, where scans from early time points cannot necessarily be registered to scans at later timepoints, even when doing intra-subject registration. This makes the strategy of registering all brains together in an iterative fashion ineffective. As noted in the Introduction, two examples of this type of study include both tumor growth and normal development. Although it is not possible to register together early and late time points in these types of studies, adjacent time points can often be accurately registered.

In order to address this type of longitudinal study, we have created the registration chain application, schematically depicted in **Figure 10**. This pipeline works as follows: Each subject is first linearly and then non-linearly registered to the next scan in the time series for that mouse. This is done first through an LSQ12 registration from source (timepoint i) to target (timepoint $i + 1$), followed immediately by a non-linear registration from source (i) to target ($i + 1$). Once this has been done for all subjects, one time point is chosen as the common time point for the registration. All scans at this timepoint are then registered together via the iterative procedure described previously. This creates the common space required for statistical analysis. The appropriate transforms from this common space to each individual scan are then concatenated and deformation fields calculated.

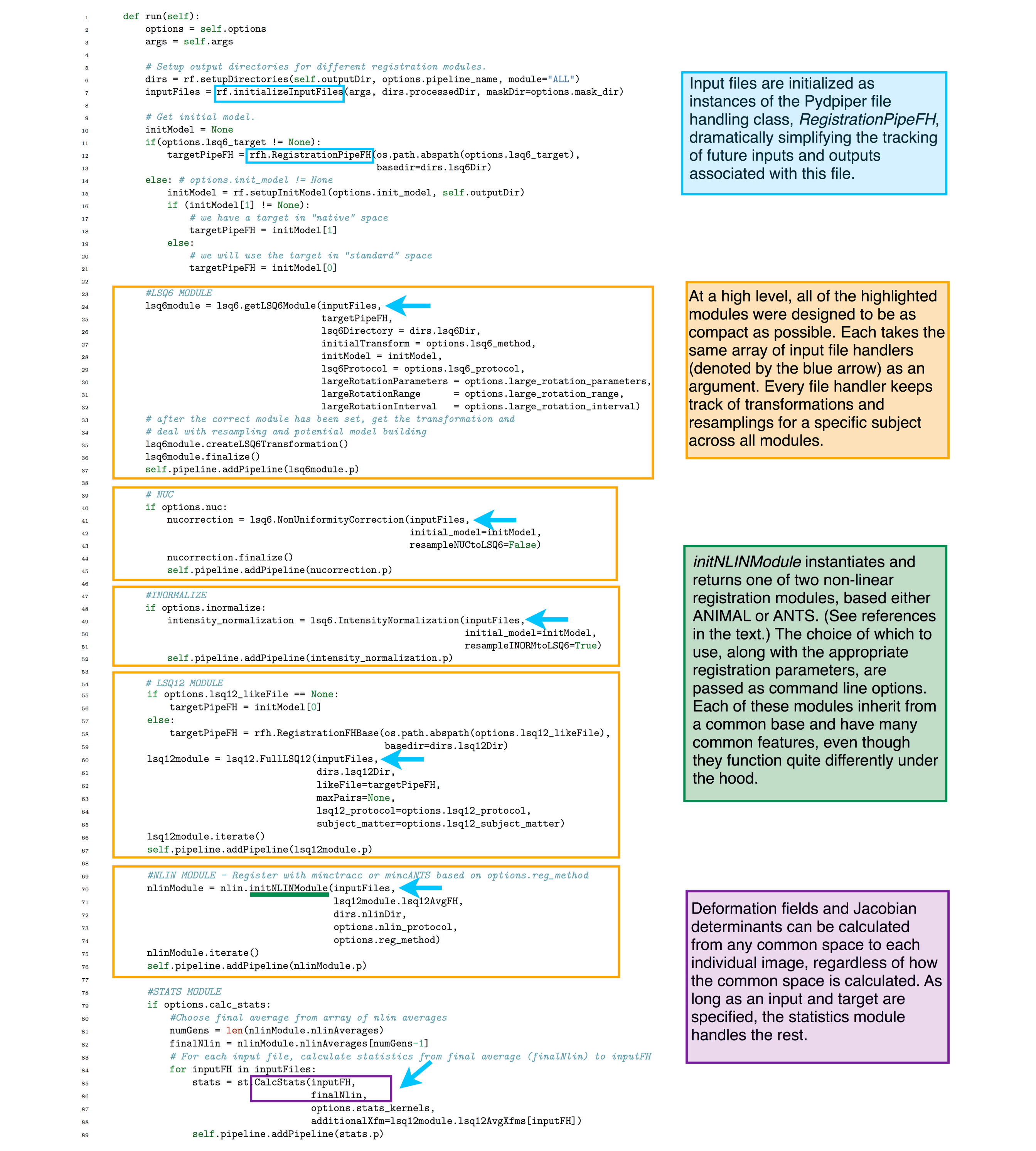
The code used to accomplish this type of registration has many parallels to the example shown in **Figure 9**. Like iterative group-wise registration, the registration chain is composed of a number of smaller modules, making the application easy to read. The main registration loop, which aligns scan i to $i + 1$ for each subject, is extremely compact: choosing minctracc results in a call to `HierarchicalMinctracc`, shown in **Figure 5**, and choosing mincANTS calls a very similar function (`LSQ12ANTSNlin`), which uses the LSQ12 module in combination with the `mincANTS` atom to appropriately align input to target. To create a common space for analysis, all subjects at a specified timepoint are then registered together using the iterative procedure described in section 4.2. Deformation fields are calculated from the common space via a subclass of the `CalcStats` class highlighted in **Figure 9**.

4.4. TWO-LEVEL REGISTRATION

Two-level registration is a registration paradigm that creates both subject and population averages. It is appropriate for data sets where all subjects are scanned multiple times, but in contrast to the types of longitudinal registration described in section 4.3, all timepoints for a given subject can be registered together. This is done using iterative group-wise registration to create a subject-specific average, enabling meaningful statistical comparison among all timepoints for a given subject. All of these subject-specific averages are then registered together, again using the iterative group-wise procedure, to create a population average. Transform concatenation can then be used to calculate the appropriate transform from the population average to each subject specific average, and subsequently to each individual scan. This allows for inter-subject comparison at each of the timepoints in the study. A schematic of this is shown in **Figure 11**.

4.5. MULTIPLE AUTOMATICALLY GENERATED TEMPLATES (MAGeT)

Of particular interest in the neuroimaging community is the ability to match MRI volumes to expertly labeled atlases, as structural segmentations are a powerful tool for enhancing analyses



```
def run(self):
    options = self.options
    args = self.args

    # Setup output directories for different registration modules.
    dirs = rf.setupDirectories(self.outputDir, options.pipeline_name, module="ALL")
    inputFiles = rf.initializeInputFiles(args, dirs.processedDir, maskDir=options.mask_dir)

    # Get initial model.
    initModel = None
    if(options.lsq6_target != None):
        targetPipeFH = rfh.RegistrationPipeFH(os.path.abspath(options.lsq6_target),
                                              basedir=dirs.lsq6Dir)
    else: # options.init_model != None
        initModel = rf.setupInitModel(options.init_model, self.outputDir)
        if (initModel[1] != None):
            # we have a target in "native" space
            targetPipeFH = initModel[1]
        else:
            # we will use the target in "standard" space
            targetPipeFH = initModel[0]

    #LSQ6 MODULE
    lsq6module = lsq6.getLSQ6Module(inputFiles,
                                    targetPipeFH,
                                    lsq6Directory = dirs.lsq6Dir,
                                    initialTransform = options.lsq6_method,
                                    initModel = initModel,
                                    lsq6Protocol = options.lsq6_protocol,
                                    largeRotationParameters = options.large_rotation_parameters,
                                    largeRotationRange      = options.large_rotation_range,
                                    largeRotationInterval   = options.large_rotation_interval)
    # after the correct module has been set, get the transformation and
    # deal with resampling and potential model building
    lsq6module.createLSQ6Transformation()
    lsq6module.finalize()
    self.pipeline.addPipeline(lsq6module.p)

    # NUC
    if options.nuc:
        nucorrection = lsq6.NonUniformityCorrection(inputFiles,
                                                    initial_model=initModel,
                                                    resampleNUCtoLSQ6=False)
        nucorrection.finalize()
        self.pipeline.addPipeline(nucorrection.p)

    #INORMALIZE
    if options.inormalize:
        intensity_normalization = lsq6.IntensityNormalization(inputFiles,
                                                              initial_model=initModel,
                                                              resampleINORMtoLSQ6=True)
        self.pipeline.addPipeline(intensity_normalization.p)

    # LSQ12 MODULE
    if options.lsq12_likeFile == None:
        targetPipeFH = initModel[0]
    else:
        targetPipeFH = rfh.RegistrationFHBase(os.path.abspath(options.lsq12_likeFile),
                                              basedir=dirs.lsq12Dir)
    lsq12module = lsq12.FullLSQ12(inputFiles,
                                  dirs.lsq12Dir,
                                  likeFile=targetPipeFH,
                                  maxPairs=None,
                                  lsq12_protocol=options.lsq12_protocol,
                                  subject_matter=options.lsq12_subject_matter)
    lsq12module.iterate()
    self.pipeline.addPipeline(lsq12module.p)

    #NLIN MODULE - Register with minctracc or mincANTS based on options.reg_method
    nlinModule = nlin.initNLINModule(inputFiles,
                                     lsq12module.lsq12AvgFH,
                                     dirs.nlinDir,
                                     options.nlin_protocol,
                                     options.reg_method)
    nlinModule.iterate()
    self.pipeline.addPipeline(nlinModule.p)

    #STATS MODULE
    if options.calc_stats:
        #Choose final average from array of nlin averages
        numGens = len(nlinModule.nlinAverages)
        finalNlin = nlinModule.nlinAverages[numGens-1]
        # For each input file, calculate statistics from final average (finalNlin) to inputFH
        for inputFH in inputFiles:
            stats = st.CalcStats(inputFH,
                                 finalNlin,
                                 options.stats_kernels,
                                 additionalXfm=lsq12module.lsq12AvgXfms[inputFH])
            self.pipeline.addPipeline(stats.p)
```

Input files are initialized as instances of the Pydpiper file handling class, *RegistrationPipeFH*, dramatically simplifying the tracking of future inputs and outputs associated with this file.

At a high level, all of the highlighted modules were designed to be as compact as possible. Each takes the same array of input file handlers (denoted by the blue arrow) as an argument. Every file handler keeps track of transformations and resamplings for a specific subject across all modules.

initNLINModule instantiates and returns one of two non-linear registration modules, based either ANIMAL or ANTS. (See references in the text.) The choice of which to use, along with the appropriate registration parameters, are passed as command line options. Each of these modules inherit from a common base and have many common features, even though they function quite differently under the hood.

Deformation fields and Jacobian determinants can be calculated from any common space to each individual image, regardless of how the common space is calculated. As long as an input and target are specified, the statistics module handles the rest.

FIGURE 9 | `run()` function in the iterative group-wise registration application. This piece of code illustrates how an extremely complex pipeline can be built up from smaller modules making it simple to read at the application level.

(Nieman et al., 2007; Dorr et al., 2008). Unfortunately, creating accurate atlases, particularly across the whole brain, can be challenging. While manual segmentation is often considered the "gold standard" for atlas creation (see e.g., Burk et al., 2004), it is too time-consuming and subjective for the ever-increasing amount of structural MRI data that must be analyzed. As such, automated atlas creation is a powerful and necessary tool and one that we wanted to include in Pydpiper.

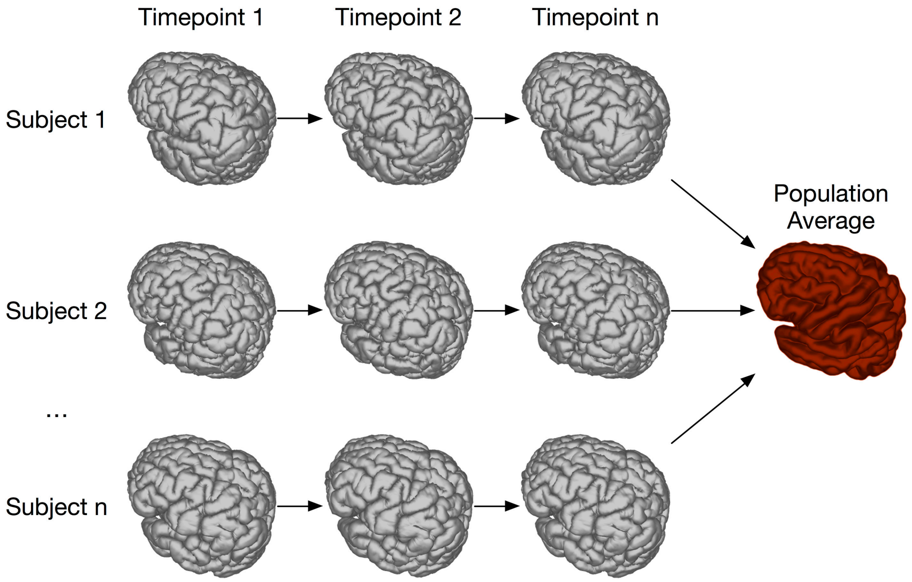


FIGURE 10 | Schematic of the registration chain pipeline. In this schematic, each subject is scanned at a series of timepoints. The arrows indicate that, within each subject, timepoint i is registered to timepoint $i + 1$. All subject scans at timepoint n are then iteratively registered together, creating a common space among all subjects and timepoints. Alternatively, a different timepoint could be chosen as the common space.

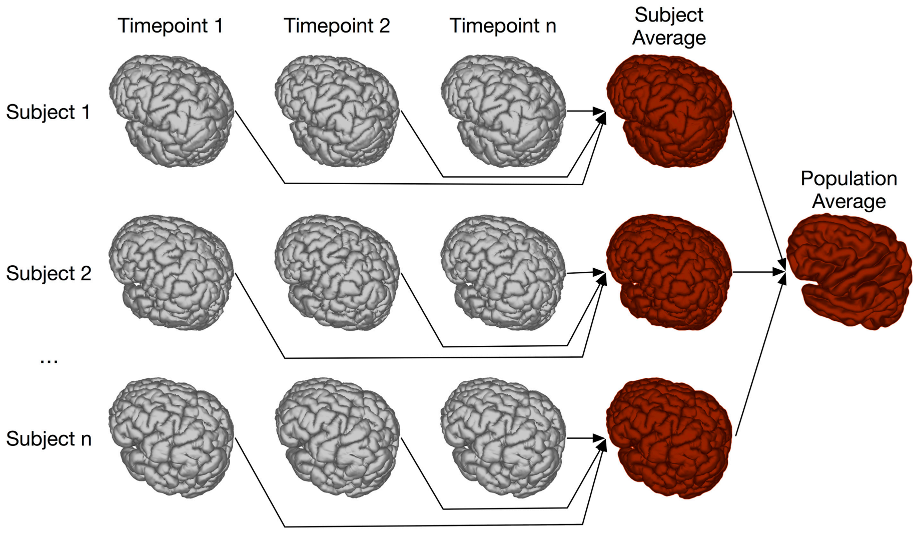


FIGURE 11 | Schematic of the two-level pipeline.

The creation of multiple automatically generated templates from a single labeled brain (MAGeT Brain), as introduced in (Chakravarty et al., 2013), is an example of a multi-atlas based, label fusion technique that produces accurate atlases without the need for manual segmentation. Briefly, it works as follows: using an input template with a set of pre-defined labels, this brain is non-linearly aligned to another subject or set of subjects. Typically, this proceeds first with an LSQ12 alignment, followed by a non-linear registration from source (template) to target (subject). The resulting transforms are then applied to the template labels, such that each subject is now labeled as well. Then, all of the subjects are non-linearly registered together (again, first with an LSQ12 alignment, followed by a non-linear registration), creating a set of labels for each subject. A label voting technique is then applied at each voxel, such that the most frequently occurring label is selected for the final segmentation of that voxel. This whole procedure is graphically depicated in **Figure 12**. We note that although MAGeT Brain was the explicit motivation for this application, the code could be easily extended to implement more sophisticated label fusion techniques (Wang et al., 2013).

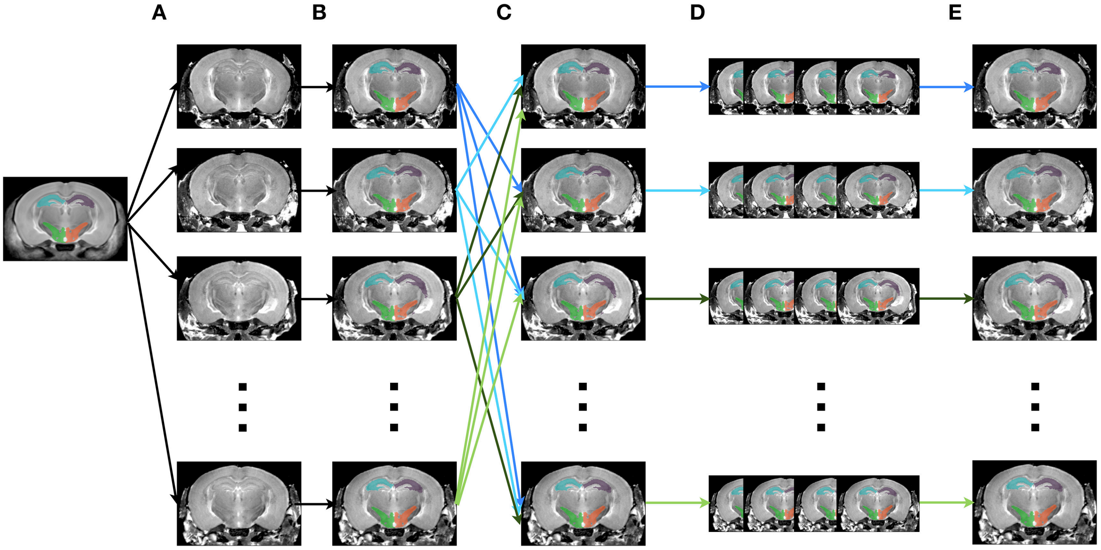


FIGURE 12 | Schematic of the MAGeT algorithm. (A) An initial labeled template is non-linearly aligned to a series of subjects. **(B)** Using the transform that results from step A, the labels from the template are propagated to each subject, creating a unique set of labels for that subject. **(C)** Each subject is non-linearly registered to every other subject. **(D)** The initial set of labels from each subject (created in step B) are propagated to every other subject using the transforms from step C. This creates a library of labels for each subject. **(E)** A voxel voting procedure is applied, creating the best set of labels for each subject.

As implemented in the Pydpiper framework, MAGeT re-uses many of the classes and modules from other applications. For example, the alignment of template to subject uses either `HierarchicalMinctracc` or `LSQ12ANTSNlin`, exactly as is done for the registration chain. This again illustrates the modular, re-usable nature of this toolkit. Prior to this alignment is the option to use the LSQ6 module for an initial alignment as well. In addition to assessing volumetric differences based on label segmentations, the Pydpiper MAGeT application can also be used in a number of different but related ways. As an example, an input template (or set of templates) can be registered to the population average created from any of the registration pipelines detailed above. After voxel voting (necessary if more than one template atlas is used), these labels from the population average can be back-propagated (via the appropriately concatenated transforms) to each individual subject in the study, enabling volumetric analysis from these sets of labels.

5. ANNOTATED CODE EXAMPLE

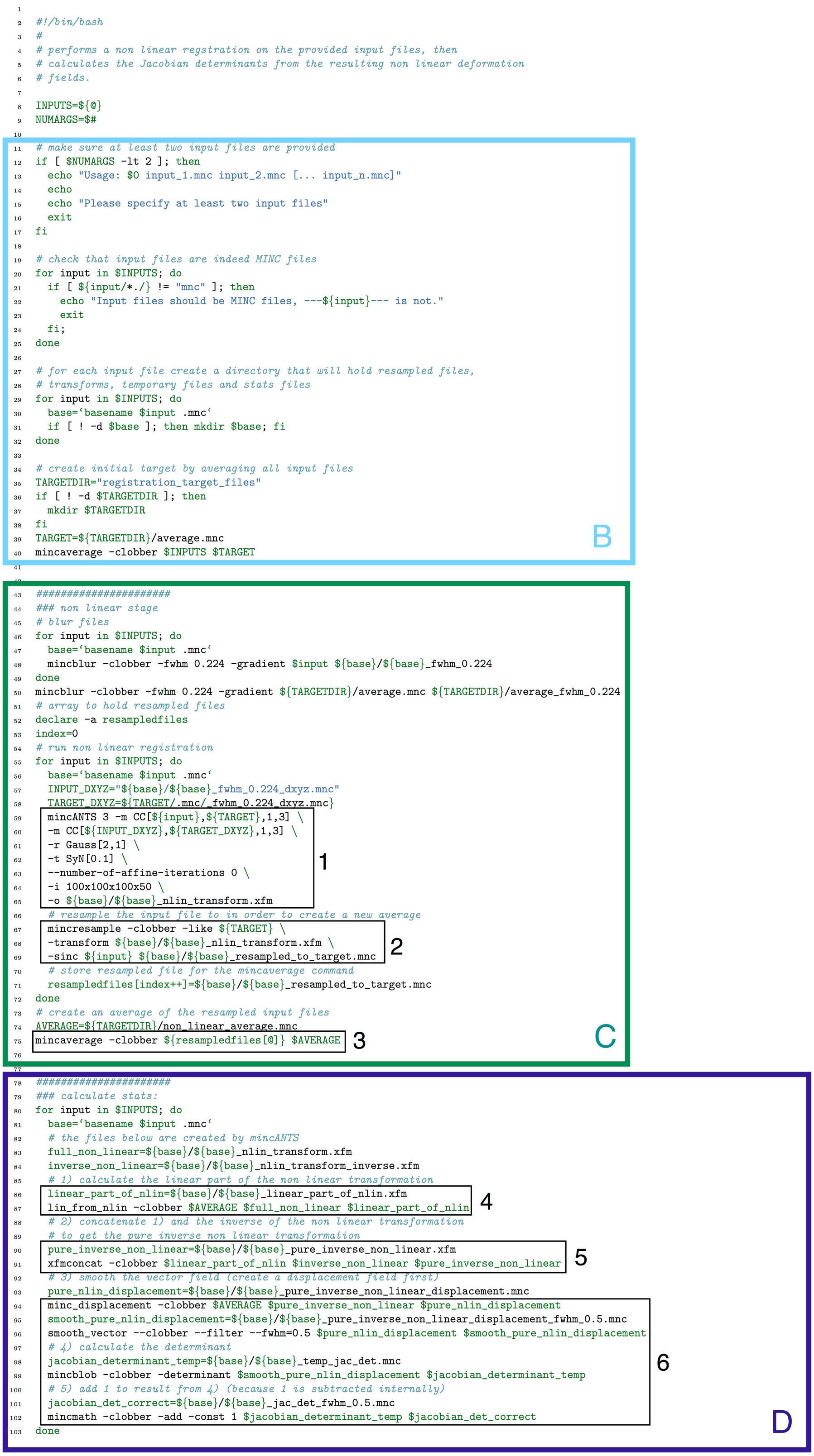
In this section, we provide a more complete Pydpiper code example along with a corresponding shell script that one might write to execute some of the same commands. These constrasting pieces of code illustrate the utility of many of the Pydpiper atoms and modules and provide a more detailed example for understanding many aspects of the code discussed throughout this paper. Additionally, because the initial Pydpiper applications are all based on the MINC file format, this section provides a bit more context regarding the command line tools we are using. For more details, we refer the reader to http://www.bic.mni.mcgill.ca/ServicesSoftware/MINC and http://en.wikibooks.org/wiki/MINC.

The example pipeline we show here corresponds to a single iteration of the multi-generation non-linear module discussed in section 4.1, followed by the calculation of the displacement field and Jacobian determinant necessary for DBM. It does the following:

1. Aligns each input subject to a specified template using mincANTS. This will result in a transform from each input to the resulting template. For clarity throughout this section, we will refer to this transform as the "final non-linear transform."
2. Resample each subject with its unique final non-linear transform.
3. Create an average of these resampled brains to create a new non-linear average.
4. Calculate the linear part of each subject's non-linear transform. The inverse of the full non-linear source-to-target transform is also needed, but is automatically calculated by mincANTS.
5. Concatenate these transforms to calculate the pure non-linear transformation from target to each individual subject.
6. Calculate the pure non-linear vector field for each subject, apply a Gaussian smoothing, and calculate the Jacobian determinant of this smoothed vector field.

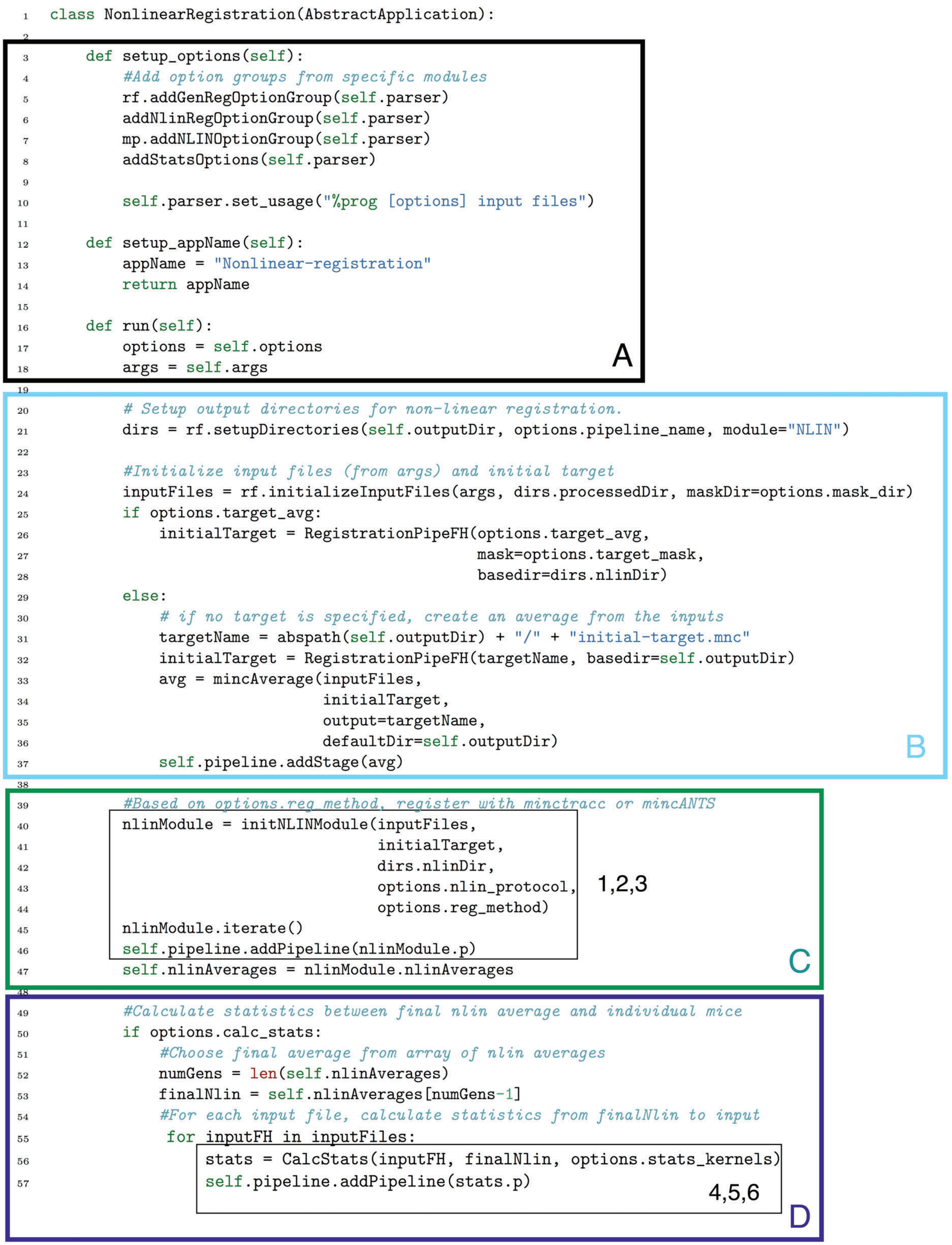
Prior to starting this registration, we make the assumption that the input files to this pipeline have already been aligned into a common space by the LSQ6 and/or LSQ12 modules described in section 4.1 of the text.

In **Figure 13** we show how the above pipeline would be executed in a simple bash script. In **Figure 14**, we show the same pipeline in Pydpiper. In this case, we show the



```
#!/bin/bash
#
# performs a non linear regstration on the provided input files, then
# calculates the Jacobian determinants from the resulting non linear deformation
# fields.

INPUTS=${@}
NUMARGS=$#

# make sure at least two input files are provided
if [ $NUMARGS -lt 2 ]; then
  echo "Usage: $0 input_1.mnc input_2.mnc [... input_n.mnc]"
  echo
  echo "Please specify at least two input files"
  exit
fi

# check that input files are indeed MINC files
for input in $INPUTS; do
  if [ ${input/*./} != "mnc" ]; then
    echo "Input files should be MINC files, ---${input}--- is not."
    exit
  fi;
done

# for each input file create a directory that will hold resampled files,
# transforms, temporary files and stats files
for input in $INPUTS; do
  base=‘basename $input .mnc‘
  if [ ! -d $base ]; then mkdir $base; fi
done

# create initial target by averaging all input files
TARGETDIR="registration_target_files"
if [ ! -d $TARGETDIR ]; then
  mkdir $TARGETDIR
fi
TARGET=${TARGETDIR}/average.mnc
mincaverage -clobber $INPUTS $TARGET


#####################
### non linear stage
# blur files
for input in $INPUTS; do
  base=‘basename $input .mnc‘
  mincblur -clobber -fwhm 0.224 -gradient $input ${base}/${base}_fwhm_0.224
done
mincblur -clobber -fwhm 0.224 -gradient ${TARGETDIR}/average.mnc ${TARGETDIR}/average_fwhm_0.224
# array to hold resampled files
declare -a resampledfiles
index=0
# run non linear registration
for input in $INPUTS; do
  base=‘basename $input .mnc‘
  INPUT_DXYZ="${base}/${base}_fwhm_0.224_dxyz.mnc"
  TARGET_DXYZ=${TARGET/.mnc/_fwhm_0.224_dxyz.mnc}
  mincANTS 3 -m CC[${input},${TARGET},1,3] \
  -m CC[${INPUT_DXYZ},${TARGET_DXYZ},1,3] \
  -r Gauss[2,1] \
  -t SyN[0.1] \
  --number-of-affine-iterations 0 \
  -i 100x100x100x50 \
  -o ${base}/${base}_nlin_transform.xfm
  # resample the input file to in order to create a new average
  mincresample -clobber -like ${TARGET} \
  -transform ${base}/${base}_nlin_transform.xfm \
  -sinc ${input} ${base}/${base}_resampled_to_target.mnc
  # store resampled file for the mincaverage command
  resampledfiles[index++]=${base}/${base}_resampled_to_target.mnc
done
# create an average of the resampled input files
AVERAGE=${TARGETDIR}/non_linear_average.mnc
mincaverage -clobber ${resampledfiles[@]} $AVERAGE


#####################
### calculate stats:
for input in $INPUTS; do
  base=‘basename $input .mnc‘
  # the files below are created by mincANTS
  full_non_linear=${base}/${base}_nlin_transform.xfm
  inverse_non_linear=${base}/${base}_nlin_transform_inverse.xfm
  # 1) calculate the linear part of the non linear transformation
  linear_part_of_nlin=${base}/${base}_linear_part_of_nlin.xfm
  lin_from_nlin -clobber $AVERAGE $full_non_linear $linear_part_of_nlin
  # 2) concatenate 1) and the inverse of the non linear transformation
  # to get the pure inverse non linear transformation
  pure_inverse_non_linear=${base}/${base}_pure_inverse_non_linear.xfm
  xfmconcat -clobber $linear_part_of_nlin $inverse_non_linear $pure_inverse_non_linear
  # 3) smooth the vector field (create a displacement field first)
  pure_nlin_displacement=${base}/${base}_pure_inverse_non_linear_displacement.mnc
  minc_displacement -clobber $AVERAGE $pure_inverse_non_linear $pure_nlin_displacement
  smooth_pure_nlin_displacement=${base}/${base}_pure_inverse_non_linear_displacement_fwhm_0.5.mnc
  smooth_vector --clobber --filter --fwhm=0.5 $pure_nlin_displacement $smooth_pure_nlin_displacement
  # 4) calculate the determinant
  jacobian_determinant_temp=${base}/${base}_temp_jac_det.mnc
  mincblob -clobber -determinant $smooth_pure_nlin_displacement $jacobian_determinant_temp
  # 5) add 1 to result from 4) (because 1 is subtracted internally)
  jacobian_det_correct=${base}/${base}_jac_det_fwhm_0.5.mnc
  mincmath -clobber -add -const 1 $jacobian_determinant_temp $jacobian_det_correct
done
```

FIGURE 13 | Bash script that does a non-linear alignment from a set of inputs to a common target, then calculates the resulting deformation fields and their Jacobian determinants. Note that our labeled sections for this figure begin with section B, as described in the text. **(B)** File checking and initialization of average; **(C)** Image alignment; **(D)** Statistics calculation.



```python
class NonlinearRegistration(AbstractApplication):

    def setup_options(self):
        #Add option groups from specific modules
        rf.addGenRegOptionGroup(self.parser)
        addNlinRegOptionGroup(self.parser)
        mp.addNLINOptionGroup(self.parser)
        addStatsOptions(self.parser)

        self.parser.set_usage("%prog [options] input files")

    def setup_appName(self):
        appName = "Nonlinear-registration"
        return appName

    def run(self):
        options = self.options
        args = self.args

        # Setup output directories for non-linear registration.
        dirs = rf.setupDirectories(self.outputDir, options.pipeline_name, module="NLIN")

        #Initialize input files (from args) and initial target
        inputFiles = rf.initializeInputFiles(args, dirs.processedDir, maskDir=options.mask_dir)
        if options.target_avg:
            initialTarget = RegistrationPipeFH(options.target_avg,
                                               mask=options.target_mask,
                                               basedir=dirs.nlinDir)
        else:
            # if no target is specified, create an average from the inputs
            targetName = abspath(self.outputDir) + "/" + "initial-target.mnc"
            initialTarget = RegistrationPipeFH(targetName, basedir=self.outputDir)
            avg = mincAverage(inputFiles,
                              initialTarget,
                              output=targetName,
                              defaultDir=self.outputDir)
            self.pipeline.addStage(avg)

        #Based on options.reg_method, register with minctracc or mincANTS
        nlinModule = initNLINModule(inputFiles,
                                    initialTarget,
                                    dirs.nlinDir,
                                    options.nlin_protocol,
                                    options.reg_method)
        nlinModule.iterate()
        self.pipeline.addPipeline(nlinModule.p)
        self.nlinAverages = nlinModule.nlinAverages

        #Calculate statistics between final nlin average and individual mice
        if options.calc_stats:
            #Choose final average from array of nlin averages
            numGens = len(self.nlinAverages)
            finalNlin = self.nlinAverages[numGens-1]
            #For each input file, calculate statistics from finalNlin to input
            for inputFH in inputFiles:
                stats = CalcStats(inputFH, finalNlin, options.stats_kernels)
                self.pipeline.addPipeline(stats.p)
```

FIGURE 14 | Non-linearRegistration application in Pydpiper. This code aligns a set of inputs toward a common target, iterating over multiple generations if requested. Note that we have omitted `if __name__ = "__main__"` from this figure, but it is included in the `.py` file that runs this code. (See **Figure 6** for more discussion). **(A)** Pre-requisites for `AbstractApplication` class and integration into pipeline; **(B)** File checking and initialization of average; **(C)** Image alignment; **(D)** Statistics calculation.

NonlinearRegistration application, which inherits from AbstractApplication and can be run on the command line. Each of these figures has multiple sections of code highlighted, and each highlighted section is labeled. The color and label of one section in **Figure 13** corresponds to the same color and label in **Figure 14**. We will use these labels as a guide for discussion. In addition, registration steps 1–6, as enumerated above, are also labeled in each figure.

5.1. SETUP AND PREREQUISITES

One of the most notable differences between the bash script in **Figure 13** and the Pydpiper code in **Figure 14** is the initial code set-up and file checking. For the bash script, this is encapsulated in section B, whereas in the Pydpiper code, this is encapsulated in sections A, B. Note that the bash script does not have section A, as it has nothing analogous to Pydpiper's AbstractApplication class.

In section A of **Figure 14**, there are two functions: setup_options and setup_appName. Both of these are necessary subclasses of AbstractApplication. setup_options adds various option groups to the application's option parser, to ensure that the appropriate command line options are available. In addition to reducing the amount of hard coding with this application, all of the command line options themselves are grouped together based on their functionality and can be reused in many different applications. setup_appName defines an application name, which is particularly useful for parsing log files. The key thing to note about section A is that, because NonlinearRegistration inherits from AbstractApplication, all of the components necessary for putting together a larger pipeline, calculating stage dependencies, and using the executor model for running multiple stages concurrently is present. No additional setup or coding is needed. In contrast, when using the simple bash script provided, stages can only be run consecutively, one-at-a-time.

In section B of both figures, three things are accomplished, albeit in quite different ways. The first is the checking that is done to ensure that all of the input files are in the MINC file format and that a minimum of two are specified. The second is that output directories are created, one for each input file. Finally, an initial target for non-linear alignment is created by averaging all of the input files.

In **Figure 13**, file checking is accomplished on lines 12–25 of code. In **Figure 14**, this happens on line 24 in the function call initializeInputFiles. Not only does this function check for the appropriate number and format of files, but it initializes each of these files as a file handler, as discussed in Section 3.2. In addition to file handler instantiation, if the options.mask_dir argument is specified, a mask will be assigned to each of the input files and their corresponding file handlers. In order to include a mask in the bash script, it would need to be re-written. In spite of the significant additional features this function adds over the corresponding bash script, it contains only 47 lines of code (not shown). Output directory creation happens on lines 29–32 of the bash script, and via two function calls in the Pydpiper code. First, on line 21, the setupDirectories function, used in virtually all other Pydpiper applications to date, creates the main output directories for the registration. Then, as part of initializeInputFiles, a subdirectory is created for each input file.

Finally, on lines 35–40, the bash script calls mincaverage to create an average target from the set of input files. This is accomplished on lines 30–36 of the Pydpiper code, though as is shown on lines 24–28, Pydpiper allows you to specify an initial target on the command line, so averaging is not always necessary. In both Pydpiper scenarios, the target file is initialized as a file handler (lines 26 or 32). Because averaging happens using the mincAverage atom (line 33), all of the appropriate file dependencies are included in the pipeline.

5.2. IMAGE ALIGNMENT

The portion of each piece of code that does image alignment is marked in both figures as section C. In **Figure 13**, a simple image alignment is shown on lines 46–65. Each input file is first blurred (lines 46–49) with the mincblur tool, using a Gaussian smoothing kernel with a full-width at half maximum (fwhm) of 0.224 μm. The target is blurred as well (line 53). Then, the blurred version of each input file is aligned to the blurred version of the target via a mincANTS call (lines 59–65). This particular call uses a cross-correlation similarity metric (CC) with a Gaussian regularizer (Gauss[2,1]) and a transformation model that uses symmetric normalization (SyN[0.1]). More details about these parameters can be found in Avants et al. (2008). The resulting transform is then applied to each of the input subjects via a mincresample call (lines 67–69) and a new average is created via mincaverage (line 75). Although this is a straightforward and brief script, it requires editing for any set of images that do not use these hard coded parameters, and extending it to multiple generations would require a fair amount of recoding.

The Pydpiper code that accomplishes this same alignment is effectively encapsulated two function calls, shown on lines 40–45 of **Figure 14**. First, the initNLINModule function is called on line 40. This function returns the appropriate non-linear module as nlinModule. The module returned depends on the value of options.reg_method passed into the function. In the example here, options.reg_method=mincANTS is specified on the command line, and initNLINModule returns an instance of NLINANTS.

After the instantiation of NLINANTS, the iterate() function is called. This function executes the following commands: After blurring both input and target using the blur atom, the blurred version of each input is registered to the blurred version of the target using the mincANTS atom. Then, as in the bash script, the resulting transform is applied to each input, and it is resampled via the mincresample atom. Then, the mincaverage atom is used to create a new non-linear average. (If additional generations were required, the new average would be blurred, and each blurred input would be registered to this new average, with the entire cycle repeating). Note that each of these atoms calls the command line tool of the same name, and the commands exectued are nearly identical (provided the same set of parameters) as those shown in the bash script.

The exact registration parameters used by NLINANTS, including (but not limited to) the Gaussian smoothing kernel necessary for blurring, the similarity metric for alignment and the transformation metric are all contained in the file specified for options.nlin_protocol (line 43). If no protocol is specified, a set of defaults, currently optimized for registration of mouse brains, is used. For the present example, the parameters necessary for only one generation are included in the protocol file. In contrast to the bash script, simply updating the non-linear protocol extends the code to an arbitrary number of generations. No re-coding is necessary.

5.3. STATISTICS CALCULATION

Finally, in section D of each figure, we show the code necessary for performing a statistics calculation. As is evident from the bash script in **Figure 13**, calculating a Jacobian determinant is a multi-step process: First, the linear part of the non-linear transform from input to target is calculated (line 87). Then, this transform is concatenated with the full transform from target to input (automatically calculated by mincANTS during the alignment procedure) via xfmconcat on line 91. After a calculation (line 94) and smoothing (line 96) of the displacement field, the Jacobian determinant is calculated (lines 99–102). Note that the determinant smoothing happens for only a single blurring kernel (in this case, the specified fwhm is 0.5 μm), and keeping track of all the inputs and outputs is a critical step in making sure this script executes properly.

In constrast, the Pydpiper execution of this code is contained entirely on line 60. For each input and target, the CalcStats class is instantiated. Within this class, fullStatsCalc executes each of the same stages as in the bash script using the appropriate atoms and modules. The deformation field may be smoothed with more than one blurring kernel (a list is specified as the --stats-kernels command line option). This list of blurs is passed as the options.stats_kernels argument to CalcStats and results in the calculation of multiple Jacobian determinant fields. Additionally, on lines 52–53, the target file necessary for the statistical calculations is selected as the final average from a series that may be generated; in the current example, this number is one, but will be larger for multi-generation registration.

Finally, we note the similarities between **Figure 14** and **Figure 9**. In particular, the code in sections C, D is nearly identical to that on lines 69–89 of **Figure 9**. This module reusibility was a deliberate design choice.

5.4. RUNNING THE CODE

To run the bash script depicted in **Figure 13**, assuming it is located in an appropriate directory in the user's path, the command is:

```
nlin_registration_and_stats.sh input_1.mnc
     input_2.mnc ... input_n.mnc
```

The analagous command for the Pydpiper code is:

```
NLIN.py input_1.mnc input_2.mnc ...
     input_n.mnc --calc-stats
--nlin-protocol=ANTS_protocol.csv
     --mask-dir=/directory/of/masks
--num-executors=1 --proc=8
```

The command line arguments for both the bash script and Pydpiper code are simply the brains to be registered (input_1.mnc ... input_n.mnc). Additional command line options are also specified for the Pydpiper code. --calc-stats is required for the final statistics calculation. (If this option is unspecified, the non-linear alignment will run but no statistics are calculated). --nlin-protocol supplies a non-linear protocol for registration, and --mask-dir specifies a directory of masks to be associated with each input. Additionally, the --num-executors and --proc options are not required, but if they are unspecified, the NLIN.py command will launch the pipeline server only, and executors will need to be launched separately.

6. DISCUSSION

The ability to use neuroimaging technologies to help understand the relationship between genotype and phenotype will be an important contribution to biomedical research in the twenty-first century. Although there are multiple different methods for analyzing neuroimaging data, image registration is of particular interest due to its wide range of applications. Performing image registration in an accurate and automated way is a critical component of of many neuroimaging studies, regardless of subject-type (humans, mice) or imaging modality (MRI, micro-CT, OPT). Different experimental designs require different registration strategies in order to assess growth patterns, compare genotype differences, or look at the impact of learning. Nevertheless, common features underlie these registration strategies, suggesting that a common computational framework may be used to construct a multitude of different registration pipelines. With the Pydpiper toolkit, we have created such a framework.

Throughout this paper, we have discussed many of the design choices that influenced our development of Pydpiper. Above all else, we were motivated by five principles: (1) high-level coding should be as simple as possible for those with less coding experience (advanced users can still easily get "under-the-hood" to create new modules); (2) individual building blocks of code should be as modular as possible, easy to subclass, and geared toward a range of biologically relevant applications; (3) complete, runnable pipelines containing thousands of stages and addressing the registration scenarios described above should be available "out-of-the-box"; (4) at the end of any pipeline, there should be an option to calculate the derived volumes necessary for TBM based statistics, using a module that contains all of the required stages; (5) we should include a robust file handling class to keep track of naming schemes and file interactions across many modules in a single application. Stemming from these principles, we believe that Pydpiper offers the following innovations to the community:

- A robust file handling class that allows access to outputs from all stages of registration at any point in the pipeline. To the best of our knowledge, no other package offers a similar framework.

- The ability to write code in a "non-linear" way; that is (as shown in **Figure 5**), duplicate stages that make conceptual sense can be written into the code, but are only executed once. This results in code that is both easy to read and write.
- A set of classes (in the form of atoms and modules) that are reusable, easy to subclass and designed to be combined in different ways to solve a variety of image registration problems.
- A toolkit that enables novice programmers to quickly piece together relatively complex pipelines with only a few lines of code.
- Four complete applications that run complex image registration pipelines with thousands of stages, "out-of-the-box."

As we noted in the Introduction and throughout the text, there are a number of pipelining frameworks currently available for running image registrations, and although our goal is not to replace any of them, we believe we offer complementary functionality. This is particularly true for Nipype, which is also open-source, written in Python, and has many of the same goals as Pydpiper. At present, Nipype offers interfaces to many more common neuroimaging toolkits than Pydpiper, and if one wanted to create a pipeline using any of these tools (e.g., FSL, Freesurfer, SPM), Nipype is the obvious choice. For other applications, such as an iterative registration using ANTs, one could choose either framework, as both Nipype and Pydpiper provide the infrastructure to do this relatively easily. Where we believe Pydpiper offers an advantage is via the integration of the file handling class into the high-level code structure. Our toolkit gives users the ability to quickly put together applications from our existing modules with relatively simple syntax, and through the file handlers, have the ability to access the state of each input at any stage throughout the pipeline. In particular, using the file handling framework in conjunction with the statistics module gives users a significant amount of flexibility in calculating statistics, making it easy to perform TBM at the end of any pipeline.

We hope that our architectural goals and code construction will attract both seasoned developers and more novice coders who want to tackle a variety of registration challenges, without having to piece together a mish-mash of functions from scratch. By creating Pydpiper as an open source, freely available toolkit, we also hope to facilitate significant additional contributions from the community. With the emergence of new imaging techniques and experimental designs will come the need for new registration paradigms, and we expect that the existing Pydpiper code provides a solid foundation on which to build these new pipelines.

ACKNOWLEDGMENTS

We gratefully acknowledge financial support from the Canadian Institutes of Health Research (CIHR) and the Ontario Brain Institute, an independent non-profit corporation, funded partially by the Ontario government. The opinions, results and conclusions are those of the authors and no endorsement by the Ontario Brain Institute is intended or should be inferred. Many of the computations necessary for code development were performed on the GPC supercomputer at the SciNet HPC Consortium (Loken et al., 2010). SciNet is funeded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontairo; Ontario Research Fund—Research Excellence; and the University of Toronto. Insights from R. Mark Henkelman, John G. Sled and Brian J. Nieman were instrumental in clarifying our thinking and we thank them for many useful discussions. Finally, we thank Fraser MacDonald for early work on this project and for coming up with the name Pydpiper.

REFERENCES

Ashburner, J., and Friston, K. J. (2000). Voxel-based morphometry–the methods. *Neuroimage* 11(6 Pt 1), 805–821. doi: 10.1006/nimg.2000.0582

Avants, B. B., Epstein, C. L., Grossman, M., and Gee, J. C. (2008). Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Med. Image Anal.* 12, 26–41. doi: 10.1016/j.media.2007.06.004

Bellec, P., Lavoie-Courchesne, S., Dickinson, P., Lerch, J. P., Zijdenbos, A. P., and Evans, A. C. (2012). The pipeline system for Octave and Matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows. *Front. Neuroinform.* 6:7. doi: 10.3389/fninf.2012.00007

Burk, K., Globas, C., Wahl, T., Buhring, U., Dietz, K., Zuhlke, C., et al. (2004). MRI-based volumetric differentiation of sporadic cerebellar ataxia. *Brain* 127(Pt 1), 175–181. doi: 10.1093/brain/awh013

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. (2006). "VisTrails: visualization meets data management," in *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data* (Chicago, IL: ACM), 745–747. doi: 10.1145/1142473.1142574

Chakravarty, M. M., Steadman, P., van Eede, M. C., Calcott, R. D., Gu, V., Shaw, P., et al. (2013). Performing label-fusion-based segmentation using multiple automatically generated templates. *Hum. Brain Mapp.* 34, 2635–2654. doi: 10.1002/hbm.22092

Chung, M. K., Worsley, K. J., Paus, T., Cherif, C., Collins, D. L., Giedd, J. N., et al. (2001). A unified statistical approach to deformation-based morphometry. *Neuroimage* 14, 595–606. doi: 10.1006/nimg.2001.0862

Collins, D. L., Holmes, C. J., Peters, T. M., and Evans, A. C. (1995). Automatic 3D model-based neuroanatomical segmentation. *Hum. Brain Mapp.* 3, 192–205. doi: 10.1002/hbm.460030304

Collins, D. L., Neelin, P., Peters, T. M., and Evans, A. C. (1994). Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space. *J. Comput. Assist. Tomogr.* 18, 192–205. doi: 10.1097/00004728-199403000-00005

Dinov, I., Lozev, K., Petrosyan, P., Liu, Z., Eggert, P., Pierce, J., et al. (2010). Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline. *PLoS ONE* 5:e13070. doi: 10.1371/journal.pone.0013070

Dorr, A. E., Lerch, J. P., Spring, S., Kabani, N., and Henkelman, R. M. (2008). High resolution three-dimensional brain atlas using an average magnetic resonance image of 40 adult c57bl/6j mice. *Neuroimage* 42, 60–69. doi: 10.1016/j.neuroimage.2008.03.037

Durrleman, S., Pennec, X., Trouvé, A., Braga, J., Gerig, G., and Ayache, N. (2013). Toward a comprehensive framework for the spatiotemporal statistical analysis of longitudinal shape data. *Int. J. Comput. Vis.* 103, 22–59. doi: 10.1007/s11263-012-0592-x

Ellegood, J., Babineau, B. A., Henkelman, R. M., Lerch, J. P., and Crawley, J. N. (2013). Neuroanatomical analysis of the BTBR mouse model of autism using magnetic resonance imaging and diffusion tensor imaging. *Neuroimage* 70, 288–300. doi: 10.1016/j.neuroimage.2012.12.029

Evans, A. C., Janke, A. L., Collins, D. L., and Baillet, S. (2012). Brain templates and atlases. *Neuroimage* 62, 911–922. doi: 10.1016/j.neuroimage.2012.01.024

Fischl, B., and Dale, A. M. (2000). Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proc. Natl. Acad. Sci. U.S.A.* 97, 11050–11055. doi: 10.1073/pnas.200033797

Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., Collins, D. L., et al. (2011). Unbiased average age-appropriate atlases for pediatric studies. *Neuroimage* 54, 313–327. doi: 10.1016/j.neuroimage.2010.07.033

Gazdzinski, L. M., and Nieman, B. J. (2014). Cellular imaging and texture analysis distinguish differences in cellular dynamics in mouse brain tumors. *Magn. Reson. Med.* 71, 1531–1541. doi: 10.1002/mrm.24790

Gogtay, N., Giedd, J. N., Lusk, L., Hayashi, K. M., Greenstein, D., Vaituzis, A. C., et al. (2004). Dynamic mapping of human cortical development during childhood through early adulthood. *Proc. Natl. Acad. Sci. U.S.A.* 101, 8174–8179. doi: 10.1073/pnas.0402680101

Good, C. D., Johnsrude, I. S., Ashburner, J., Henson, R. N., Friston, K. J., and Frackowiak, R. S. (2001). A voxel-based morphometric study of ageing in 465 normal adult human brains. *Neuroimage* 14(1 Pt 1), 21–36. doi: 10.1006/nimg.2001.0786

Gorgolewski, K., Burns, C. D., Madison, C., Clark, D., Halchenko, Y. O., Waskom, M. L., et al. (2011). Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. *Front. Neuroinform.* 5:13. doi: 10.3389/fninf.2011.00013

Guimond, A., Meunier, J., and Thirion, J.-P. (2000). Average brain models: a convergence study. *Comp. Vis. Image Understand.* 77, 192–210. doi: 10.1006/cviu.1999.0815

Hanke, M., and Halchenko, Y. O. (2011). Neuroscience runs on GNU/Linux. *Front. Neuroinform.* 5:8. doi: 10.3389/fninf.2011.00008

Heckemann, R., Hajnal, J., Aljabar, P., Rueckert, D., and Hammers, A. (2006). Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. *Neuroimage* 33, 115–126. doi: 10.1016/j.neuroimage.2006.05.061

Henkelman, R. M. (2010). Systems biology through mouse imaging centers: experience and new directions. *Ann. Rev. Biomed. Eng.* 12, 143–166. doi: 10.1146/annurev-bioeng-070909-105343

Hyde, K. L., Lerch, J., Norton, A., Forgeard, M., Winner, E., Evans, A. C., et al. (2009). Musical training shapes structural brain development. *J. Neurosci.* 29, 3019–3025. doi: 10.1523/JNEUROSCI.5118-08.2009

Ince, D. C., Hatton, L., and Graham-Cumming, J. (2012). The case for open computer programs. *Nature* 482, 485–488. doi: 10.1038/nature10836

Joshi, A. A., Shattuck, D. W., Thompson, P. M., and Leahy, R. M. (2007). Surface-Constrained Volumetric Brain Registration Using Harmonic Mappings. *IEEE Trans. Med. Imaging* 26, 1657–1669. doi: 10.1109/TMI.2007.901432

Joshi, S. H., Cabeen, R. P., Joshi, A. A., Sun, B., Dinov, I., Narr, K. L., et al. (2012). Diffeomorphic sulcal shape analysis on the cortex. *IEEE Trans. Med. Imaging* 31, 1195–1212. doi: 10.1109/TMI.2012.2186975

Kim, J. S., Singh, V., Lee, J. K., Lerch, J., Ad-Dab'bagh, Y., MacDonald, D., et al. (2005). Automated 3-D extraction and evaluation of the inner and outer cortical surfaces using a Laplacian map and partial volume effect classification. *Neuroimage* 27, 210–221. doi: 10.1016/j.neuroimage.2005.03.036

Klein, A., Andersson, J., Ardekani, B. A., Ashburner, J., Avants, B., Chiang, M. C., et al. (2009). Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* 46, 786–802. doi: 10.1016/j.neuroimage.2008.12.037

Kovačević, N., Henderson, J. T., Chan, E., Lifshitz, N., Bishop, J., Evans, A. C., et al. (2005). A three-dimensional MRI atlas of the mouse brain with estimates of the average and variability. *Cereb. Cortex* 15, 639–645. doi: 10.1093/cercor/bhh165

Lau, J. C., Lerch, J. P., Sled, J. G., Henkelman, R. M., Evans, A. C., and Bedell, B. J. (2008). Longitudinal neuroanatomical changes determined by deformation-based morphometry in a mouse model of Alzheimer's disease. *Neuroimage* 42, 19–27. doi: 10.1016/j.neuroimage.2008.04.252

Lepore, N., Brun, C. A., Chiang, M. C., Chou, Y. Y., Dutton, R. A., Hayashi, K. M., et al. (2006). Multivariate statistics of the Jacobian matrices in tensor based morphometry and their application to HIV/AIDS. *Med. Image Comput. Comput. Assist. Interv.* 9(Pt 1), 191–198. doi: 10.1007/11866565_24

Lerch, J. P., Carroll, J. B., Spring, S., Bertram, L. N., Schwab, C., Hayden, M. R., et al. (2008). Automated deformation analysis in the YAC128 Huntington disease mouse model. *Neuroimage* 39, 32–39. doi: 10.1016/j.neuroimage.2007.08.033

Lerch, J. P., Sled, J. G., and Henkelman, R. M. (2011). MRI phenotyping of genetically altered mice. *Methods Mol. Biol.* 711, 349–361. doi: 10.1007/978-1-61737-992-5-17

Loken, C., Gruner, D., Groer, L., Peltier, R., Bunn, N., Craig, M., et al. (2010). Scinet: lessons learned from building a power-efficient top-20 system and data centre. *J. Phys. Conf. Ser.* 256:012026. doi: 10.1088/1742-6596/256/1/012026

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., et al. (2006). Scientific workflow management and the Kepler system. *Concurr. Comput. Prac. Exp.* 18, 1039–1065. doi: 10.1002/cpe.994

Macdonald, D. (2000). Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *Neuroimage* 12, 340–356. doi: 10.1006/nimg.1999.0534

Maheswaran, S., Barjat, H., Bate, S. T., Aljabar, P., Hill, D. L. G., Tilling, L., et al. (2009). Analysis of serial magnetic resonance images of mouse brains using image registration. *Neuroimage* 44, 692–700. doi: 10.1016/j.neuroimage.2008.10.016

Mangin, J.-F., Jouvent, E., and Cachia, A. (2010). *In-vivo* measurement of cortical morphology: means and meanings. *Curr. Opin. Neurol.* 23, 359–367. doi: 10.1097/WCO.0b013e32833a0afc

Mazziotta, J., Toga, A., Evans, A., Fox, P., Lancaster, J., Zilles, K., et al. (2001). A probabilistic atlas and reference system for the human brain: international consortium for brain mapping (ICBM). *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 356, 1293–1322. doi: 10.1098/rstb.2001.0915

Nieman, B. J., Bishop, J., Dazai, J., Bock, N. A., Lerch, J. P., Feintuch, A., et al. (2007). Mr technology for biological studies in mice. *NMR Biomed.* 20, 291–303. doi: 10.1002/nbm.1142

Oinn, T., Greenwood, M., Addis, M., Alpdemir, M. N., Ferris, J., Glover, K., et al. (2006). Taverna: lessons in creating a workflow environment for the life sciences. *Concurr. Comput. Prac. Exp.* 18, 1067–1100. doi: 10.1002/cpe.993

Paus, T. (2010). Population neuroscience: why and how. *Hum. Brain Mapp.* 31, 891–903. doi: 10.1002/hbm.21069

Sled, J. G., Zijdenbos, A. P., and Evans, A. C. (1998). A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE Trans. Med. Imaging* 17, 87–97. doi: 10.1109/42.668698

Spring, S., Lerch, J. P., and Henkelman, R. M. (2007). Sexual dimorphism revealed in the structure of the mouse brain using three-dimensional magnetic resonance imaging. *Neuroimage* 35, 1424–1433. doi: 10.1016/j.neuroimage.2007.02.023

Studholme, C. (2011). Mapping fetal brain development *in utero* using magnetic resonance imaging: the Big Bang of brain mapping. *Annu. Rev. Biomed. Eng.* 13, 345–368. doi: 10.1146/annurev-bioeng-071910-124654

Szulc, K. U., Nieman, B. J., Houston, E. J., Bartelle, B. B., Lerch, J. P., Joyner, A. L., et al. (2013). MRI analysis of cerebellar and vestibular developmental phenotypes in Gbx2 conditional knockout mice. *Magn. Reson. Med*. 70, 1707–1717. doi: 10.1002/mrm.24597

van Eede, M. C., Scholz, J., Chakravarty, M. M., Henkelman, R. M., and Lerch, J. P. (2013). Mapping registration sensitivity in MR mouse brain images. *Neuroimage* 82, 226–236. doi: 10.1016/j.neuroimage.2013.06.004

Wang, H., Suh, J. W., Das, S. R., Pluta, J., Craige, C., and Yushkevich, P. A. (2013). Multi-atlas segmentation with joint label fusion. *IEEE Trans. Patt. Anal. Mach. Intell.* 35, 611–623. doi: 10.1109/TPAMI.2012.143

Woods, R. P., Grafton, S. T., Holmes, C. J., Cherry, S. R., and Mazziotta, J. C. (1998a). Automated image registration: I. General methods and intrasubject, intramodality validation. *J. Comput. Assist. Tomogr.* 22, 139–152. doi: 10.1097/00004728-199801000-00027

Woods, R. P., Grafton, S. T., Watson, J. D., Sicotte, N. L., and Mazziotta, J. C. (1998b). Automated image registration: II. Intersubject validation of linear and nonlinear models. *J. Comput. Assist. Tomogr.* 22, 153–165. doi: 10.1097/00004728-199801000-00028

Zijdenbos, A. P., Dawant, B. M., and Margolin, R. A. (1995). "Intensity correction and its effects on measurement variability in MRI," in *International Symposium on Computer and Communication Systems for Image Guided Diagnosis and Therapy (CAR 95)*, eds H. U. Lemke, K. Inamura, C. C. Jaffe, and M. W. Vannier (Berlin: Springer-Verlag Berlin), 216–221.

Zijdenbos, A. P., Forghani, R., and Evans, A. C. (2002). Automatic "pipeline" analysis of 3-D MRI data for clinical trials: application to multiple sclerosis. *IEEE Trans. Med. Imaging* 21, 1280–1291. doi: 10.1109/TMI.2002.806283

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 20 December 2013; accepted: 26 June 2014; published online: 30 July 2014.
Citation: Friedel M, van Eede MC, Pipitone J, Chakravarty MM and Lerch JP (2014) Pydpiper: a flexible toolkit for constructing novel registration pipelines. Front. Neuroinform. ***8****:67. doi: 10.3389/fninf.2014.00067*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

METHODS ARTICLE
published: 19 March 2014
doi: 10.3389/fninf.2014.00026

QSpike tools: a generic framework for parallel batch preprocessing of extracellular neuronal signals recorded by substrate microelectrode arrays

Mufti Mahmud[1,2], Rocco Pulizzi[1], Eleni Vasilaki[1,3] and Michele Giugliano[1,3,4]*

[1] Theoretical Neurobiology and Neuroengineering Lab, Department of Biomedical Sciences, University of Antwerp, Wilrijk, Belgium
[2] Institute of Information Technology, Jahangirnagar University, Savar, Bangladesh
[3] Department of Computer Science, University of Sheffield, Sheffield, UK
[4] Brain Mind Institute, Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Michela Chiappalone, Italian Institute of Technology, Italy
Leslie Samuel Smith, University of Stirling, UK

Correspondence:
Michele Giugliano, Theoretical Neurobiology and Neuroengineering Lab, Department of Biomedical Sciences, University of Antwerp, Universiteitsplein 1, B-2610 Wilrijk, Belgium
e-mail: michele.giugliano@uantwerpen.be

Micro-Electrode Arrays (MEAs) have emerged as a mature technique to investigate brain (dys)functions *in vivo* and in *in vitro* animal models. Often referred to as "smart" Petri dishes, MEAs have demonstrated a great potential particularly for medium-throughput studies *in vitro*, both in academic and pharmaceutical industrial contexts. Enabling rapid comparison of ionic/pharmacological/genetic manipulations with control conditions, MEAs are employed to screen compounds by monitoring non-invasively the spontaneous and evoked neuronal electrical activity in longitudinal studies, with relatively inexpensive equipment. However, in order to acquire sufficient statistical significance, recordings last up to tens of minutes and generate large amount of raw data (e.g., 60 channels/MEA, 16 bits A/D conversion, 20 kHz sampling rate: approximately 8 GB/MEA,h uncompressed). Thus, when the experimental conditions to be tested are numerous, the availability of fast, standardized, and automated signal preprocessing becomes pivotal for any subsequent analysis and data archiving. To this aim, we developed an in-house cloud-computing system, named QSpike Tools, where CPU-intensive operations, required for preprocessing of each recorded channel (e.g., filtering, multi-unit activity detection, spike-sorting, etc.), are decomposed and batch-queued to a multi-core architecture or to a computers cluster. With the commercial availability of new and inexpensive high-density MEAs, we believe that disseminating QSpike Tools might facilitate its wide adoption and customization, and inspire the creation of community-supported cloud-computing facilities for MEAs users.

Keywords: substrate arrays of microelectrodes, MEAs, extracellular, batch analysis, embarrassingly parallel signal-processing, cellular electrophysiology

INTRODUCTION

Among the most challenging open questions in Systems Neuroscience, structure-function relationship has raised a renewed interest. While novel ultrastructural anatomical investigations (Briggman and Denk, 2006; Mikula et al., 2012) promise to revolutionize the field, significant new progresses in our understanding of neuronal networks physiology and in preclinical neurotechnological applications, have been achieved by extracellularly monitoring the electrical activity of large neuronal ensembles (Rutten, 2002; Buzsaki, 2004; Schwartz, 2004; Wise et al., 2004; Lebedev and Nicolelis, 2006; Nicolelis and Lebedev, 2009). Complementary to high-resolution patch-clamp microscopic access and to mesoscopic non-invasive electroencephalography and functional magnetic resonance imaging, the extracellular interfacing of neurons to artificial devices has taken a considerable leap forward (Fromherz, 2006; Vassanelli et al., 2012; Spira and Hai, 2013).

Since its early introduction, extracellular recordings have been widely adopted both in academic and industrial pharmaceutical contexts, for monitoring and evoking neuronal activity *in vivo* and *ex vivo* under a variety of scientific, technological, neuroprosthetic, and clinical perspectives (Berdondini et al., 2006, 2009; Giugliano et al., 2008; Kim et al., 2009; Wang et al., 2012; Gortz et al., 2013; Liu et al., 2013). In addition, recent advances in real-time computing and in micro- and nanotechnologies opened brand new possibilities (Arsiero et al., 2007; Mazzatenta et al., 2007; Jain and Muthuswamy, 2008; Chen et al., 2009; Kim et al., 2009; Fendyur et al., 2011; Hai and Spira, 2012; Tian and Lieber, 2013).

However, in terms of data collection, analysis, and interpretation, multi-site extracellular recordings pose some challenges, given the large size of the raw data files acquired and the inherent complexity behind their rapid and accurate interpretation (Buzsaki, 2004; Stevenson and Kording, 2011). From Neuroinformatics perspectives, several open-source software toolboxes have been developed and released to the community over the years, addressing those issues and ultimately aimed at making electrophysiological data handling and analysis easier, faster, interactive, and more user friendly (Egert et al., 2002; Quiroga et al., 2004; Vato et al., 2004; Bonomini et al., 2005;

Wagenaar et al., 2005; Morup et al., 2007; Cui et al., 2008; Huang et al., 2008; Magri et al., 2009; Novellino et al., 2009; Bologna et al., 2010; Abdoun et al., 2011; Kwon et al., 2012; Mahmud et al., 2012; Just et al., 2013). Hardware based techniques have been also made available to the community to perform spike detection and sorting (Yu et al., 2012; Hwang et al., 2013). Nonetheless, signal processing and data analysis remain intensive, even though modern personal computing power increased dramatically and costs steadily decreased. In such perspectives, the advantages of distributed data-analysis, cloud-based computing, computer clusters, and parallel graphical co-processing have become obvious to the neuroscience community (Wilson and Williams, 2009; Chen et al., 2011, 2013a,b), in analogy to what the field is witnessing for Computational Neuroscience applications.

In this work, we addressed some of the basic requirements of substrate-integrated microelectrode arrays (MEAs) users, focusing on routine multichannel data analysis in *in vitro* studies, where several experimental conditions are examined and several binary raw data files are collected daily. We defined two major objectives that we consider a priority in our own laboratory: (i) increasing experimental throughput by freeing the data-acquisition computers from the burden of subsequent raw-signal analysis; (ii) providing the end-user with software tools that could be employed with neither previous training nor computer proficiency, but still easily customizable to include any analysis algorithm. To this aim, we developed and implemented a web-based workflow, named QSpike Tools, for the unsupervised execution of generic signal preprocessing and analysis of multichannel extracellular signals (see **Figure 1**). As sample data processing primitives for demonstration purposes, we chose a minimal set of basic operations that are performed in any multi-unit activity analysis: filtering, peak-detection, sorting, and simple spike-rate analysis. Tedious and long interactive analysis sessions could be then replaced by an automated procedure, and most important for us, by a more rational and efficient use of the existing shared computing resources in our laboratory. This was accomplished by delegating and batch queuing the preprocessing of the raw data files to an in-house multicore server. This is controlled and monitored remotely via a simple web browser, with no (computing) programming familiarity required, and leaving part of the resources of the server free for other users. Our generic framework might be successfully applied, or easily customized to include additional analysis scripts (e.g., in MATLAB), in the context of routine compounds screening, with highly consolidated analysis methods and with a set of established performance indicators. We also ultimately aimed at a scenario where no further manipulation of post-processed data may be required to the end-user, with one of the outcomes of the automated analysis being a portable document format (PDF) report, containing textual information as well as automatically generated tables, graph, and plots (see the Supplementary Material).

We are convinced by the importance of disseminating QSpike Tools to the community, as a generic, easily customizable, processing workflow, for the sake of its potential wide adoption. Indeed, robust open-source distributed (grid) platforms are often in use in many laboratories or (super)computing departmental facilities. Finally, inspired by the recent creation of community-supported Neuroinformatics shared facilities for numerical simulations, such as the NSF-funded Neuroscience Gateway Portal (NSG[1]), our work could lead to the creation of institutional or international facilities for remote automated MEA data analysis.

MATERIALS AND METHODS

MULTI-ELECTRODE ARRAY RECORDINGS OF NEURONAL MULTIUNIT ACTIVITY

Commercial microelectrode arrays (MEAs) for *in vitro* electrophysiology were obtained from Multichannel Systems (Reutlingen, Germany) and employed in routine experiments (**Figure 1**). Briefly, MEAs consists of 60 Indium Tin Oxide (ITO) planar microelectrodes (30 μm in diameter, 200 μm in spacing) with 8 by 8 regular layout, microfabricated on a glass substrate by photolithography, reactive ion etching, and physical vapor deposition. Prior to cell seeding, MEAs were autoclaved and coated by Polyethyleneimine (0.1% PEI, Sigma-Aldrich). Primary cortical cell cultures were obtained by standard methods from newborn Wistar rats (Charles River, France), following national and institutional guidelines on animal experimentation, upon enzymatic (0.025% trypsin) and mechanical dissociation. Prior to seeding, cells were centrifuged and suspended in a medium containing Modified Essential Medium supplemented with 2 M glucose, 200 mM l-glutamine, 50 μg/mL gentamycin and 5% horse serum (Sigma-Aldrich). Approximately 2000–3000 cells/mm^2 were plated on the inner area of each MEAs and maintained in culture medium (changed three times per week), at 100% relative humidity, 37°C and 5% CO_2 for 1–30 days *in vitro* (DIV). MEAs were sealed by fluorinated Teflon membranes, allowing gas but no water exchanges, reducing osmolarity alterations and contamination risks (Potter and DeMarse, 2001), making possible to perform the recordings in a low-humidity, electronic-friendly, conditions at 37°C, 5% CO_2.

The MEA microelectrodes were then employed to monitor non-invasively the collective electrical activity of neuronal networks developing *ex vivo* on their substrates. Recordings took place after 21 DIV upon mounting of each MEA into the recording amplifier (**Figure 1A**, 1060BC, Multichannel Systems, Reutlingen, Germany) and acquiring 15–30 min of spontaneous electrical activity was acquired at 25 kHz/channel, after 1200x amplification. MC Rack software (Multichannel Systems, Reutlingen, Germany) was employed to store the digitized data on disk, as multiplexed binary files (*.mcd file format), with each file containing raw voltage waveforms from all the MEA microelectrodes.

Additional hardware (**Figure 1A**) included an acquisition computer with a PCI analog to digital board (MC Card, 64 channels A/D, 4 DIO, 16bits Multichannel Systems, Reutlingen, Germany), as well as a temperature regulator and a stimulus isolator (STG1002, Multichannel Systems, Reutlingen, Germany). **Figure 1B** depicts a magnification of the inner area of a MEA

[1] http://www.nsgportal.org/

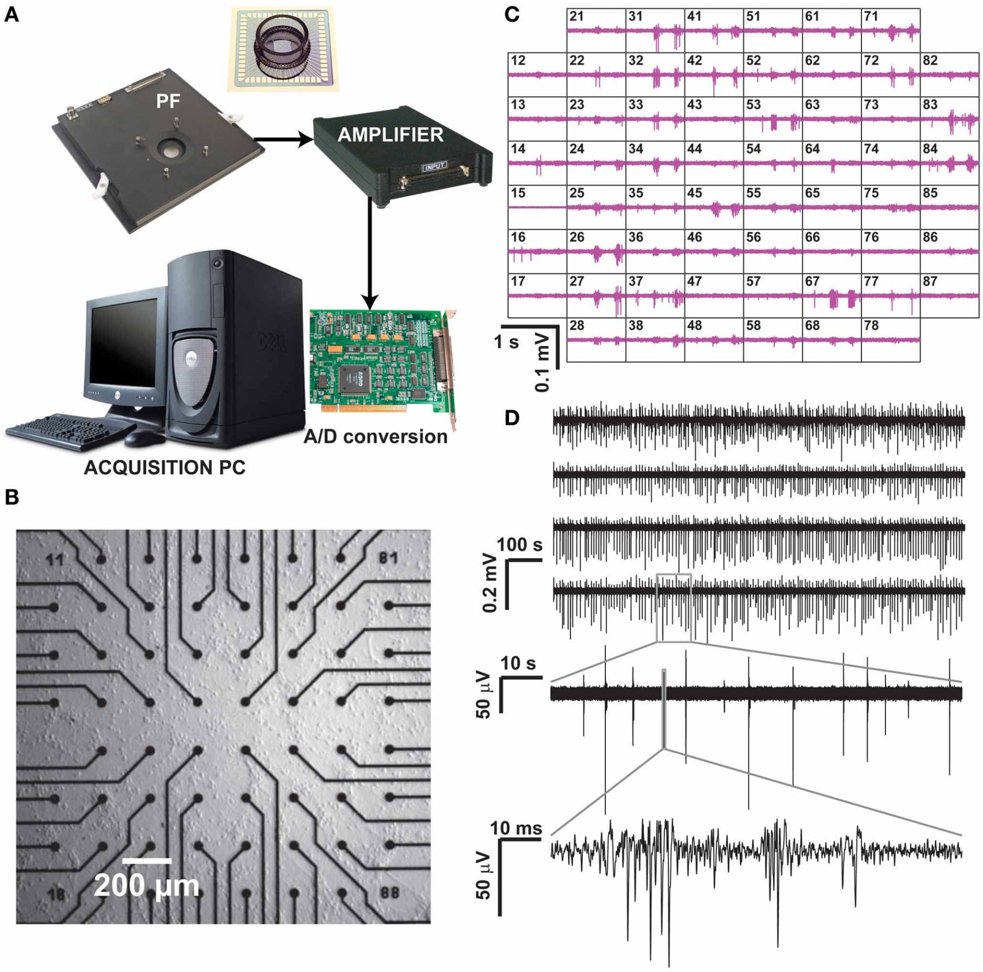


FIGURE 1 | Recording of neuronal activity *ex vivo,* by means of a commercial MEA hardware platform. Our experimental setup **(A)** is composed by a pre-amplification and filtering (PF) stage and by an additional signal amplifier, connected via an A/D board to a dedicated data acquisition PC, which also controls a temperature controller and electrical stimulus generation (not shown). After plating and culturing mammalian primary cortical neurons *ex vivo* on a MEA for several days, spontaneous electrical activity is detected and recorded at each microelectrode **(B)**, arranged as a regular 8 by 8 layout, 200 μm spacing, and displayed in real time **(C)**. Representative raw voltage traces from four sample microelectrodes, recorded over 20 min, are sown in **(D)** with increasing levels of magnification, to reveal the stereotypical pattern of spontaneous multi-unit electrical activity.

(i.e., 1 × 1 mm^2) populated by microelectrodes, and **Figure 1C** displays a typical recording session where the extracellular electrical activity sensed at each microelectrode can be monitored over time as an electrical potential. **Figure 1D** reports representative raw (off-line band-pass filtered) sample recordings, acquired over 20 min from six sample microelectrodes.

SYSTEM ARCHITECTURE

The QSpike Tools workflow is based on a client-server architecture (**Figure 2A**). The server is a stand-alone (powerful) computer workstation, or it is the master-node of a computers cluster, running a standard distribution of the Linux operating system. Accordingly, the individual processor cores of the server, or the computers of the cluster are configured as distributed computing nodes, as in a high-performance computing intranet. The master node also runs a web server software, capable of launching a series of server-side operations (e.g., via the common gateway interface, CGI[2], as in web applications) when instructed by a client computer connected to the same intranet.

The preprocessing performed by QSpike Tools includes five steps (**Figure 2B**). Some of them occur and progress automatically in a sequence, while others are only initiated via user interaction with the web page hosted by the master node upon selection appropriate hyperlinks. The links provide:

(1) A fast and secure SFTP [3] raw data transfer, from the data acquisition setup storage hard drive to the computer(s) dedicated for data preprocessing;
(2) The conversion of each multiplexed raw data file into several binary files, each containing data points from a distinct recording channel;

[2] http://en.wikipedia.org/wiki/Common_Gateway_Interface
[3] http://www.openssh.com

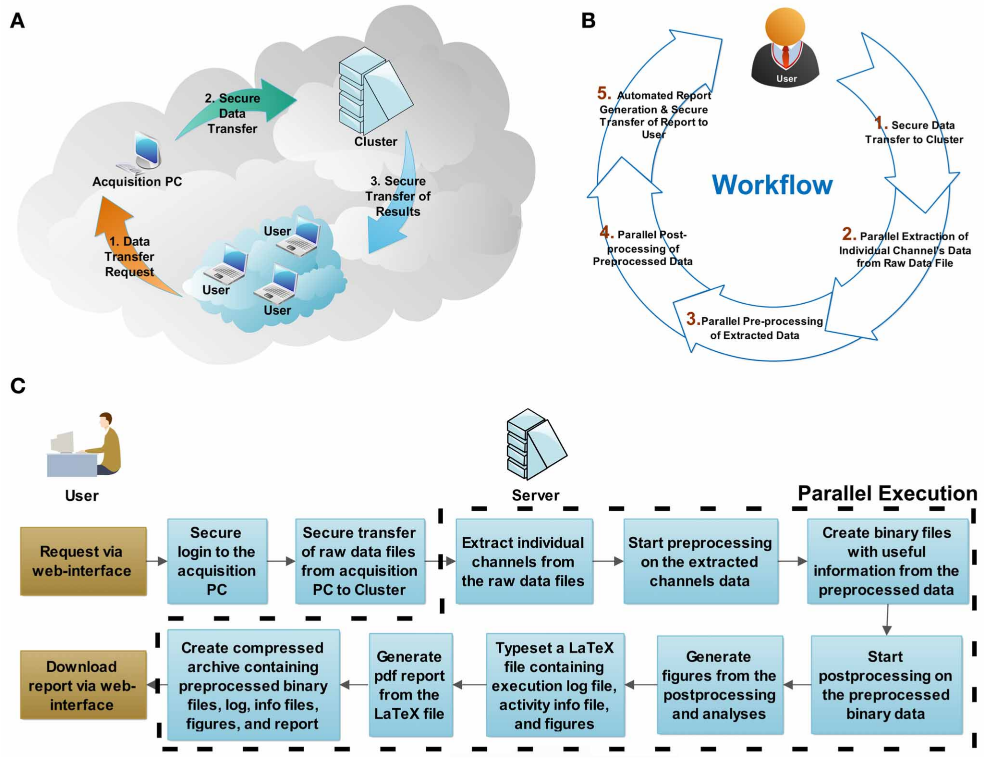

A
2. Secure Data Transfer
Cluster
Acquisition PC
3. Secure Transfer of Results
1. Data Transfer Request
User
User
User
B
User
5. Automated Report Generation & Secure Transfer of Report to User
1. Secure Data Transfer to Cluster
Workflow
4. Parallel Post-processing of Preprocessed Data
2. Parallel Extraction of Individual Channel's Data from Raw Data File
3. Parallel Pre-processing of Extracted Data
C
User
Server
Parallel Execution
Request via web-interface
Secure login to the acquisition PC
Secure transfer of raw data files from acquisition PC to Cluster
Extract individual channels from the raw data files
Start preprocessing on the extracted channels data
Create binary files with useful information from the preprocessed data
Start postprocessing on the preprocessed binary data
Generate figures from the postprocessing and analyses
Typeset a LaTeX file containing execution log file, activity info file, and figures
Generate pdf report from the LaTeX file
Create compressed archive containing preprocessed binary files, log, info files, figures, and report
Download report via web-interface

FIGURE 2 | System architecture and workflow description. The physical configuration of computers within network(s) is reminiscent **(A)** of a cloud-computing architecture. Through a web browser, the user initiates **(B)** the workflow by requesting secure data transfer from the acquisition computer to the server, and by starting the parallel batch processing and analysis. At the end of the analysis, the server generates automatically a portable document format (PDF) document, reporting on the execution log, the activity summary, and the result summary, which is incorporated in a compressed file archive along with the analysis result. When ready, the user concludes the session by securely downloading the compressed file archive to his/her personal computer. The flowchart **(C)** shows the actions required by the user and the processing steps executed by the server (cluster). The steps executed in parallel are outlined by the dashed line.

(3) The scripting-based (i.e., written in MATLAB, The Mathworks, Natick, USA), fully extensible, preprocessing sequence for each file, currently including band-pass filtering, stimulation-artifacts removal, multi-unit activity detection and elementary spike-sorting (**Figure 3B**);
(4) The additional scripting-based (i.e., MATLAB) visualization of the (multi)unit activity extracted by the previous steps (e.g., MEA-wide synchronous bursting rate, single-channels and MEA-wide firing rate, intra-burst instantaneous discharge probability, etc.);
(5) The automated typesetting of a PDF report from a dynamically generated and compiled LaTeX[4] source file, including both textual and graphical information, extracted by the previous step and secure download of the PDF report and of all intermediate and final preprocessed files (e.g., spike timestamps, spike waveforms, spike count) to the user's personal computer, as a compressed file archive.

Provided that certain dependencies are respected (e.g., step 4 must follow the completion of step 3, across all electrodes of the same data file), all the remaining steps (i.e., 2–5) can also be batch-queued and executed in parallel (e.g., one task per core, processor, or intranet node). A flowchart explaining the detailed processing steps is shown in **Figure 2C**.

The QSpike Tools workflow is in fact based on the observation that any CPU-intensive preprocessing needed can be executed in parallel, independently of any other recorded channel (Denker et al., 2010). All operations, necessary for any subsequent data analysis, can be performed in parallel across channels thus directly exploiting the advantages of embarrassingly parallel scheduling (**Figure 3**).

It is worth to note that, in our context, parallel processing implies performing pre-, post-processing and analyses on each recorded channel, independently. This excludes sophisticated spike detection, sorting, and analyses algorithms that are useful when employing, e.g., tetrode-like arranged MEAs, whose inter-electrode distances are much smaller than those employed here. In those circumstances, correlated information from distinct channels cannot be treated independently and must be jointly analyzed. While the series of analyses of QSpike Tools can be extended to include similar analysis algorithms, this has not been yet implemented in its current version as it implies a

[4] http://www.latex-project.org

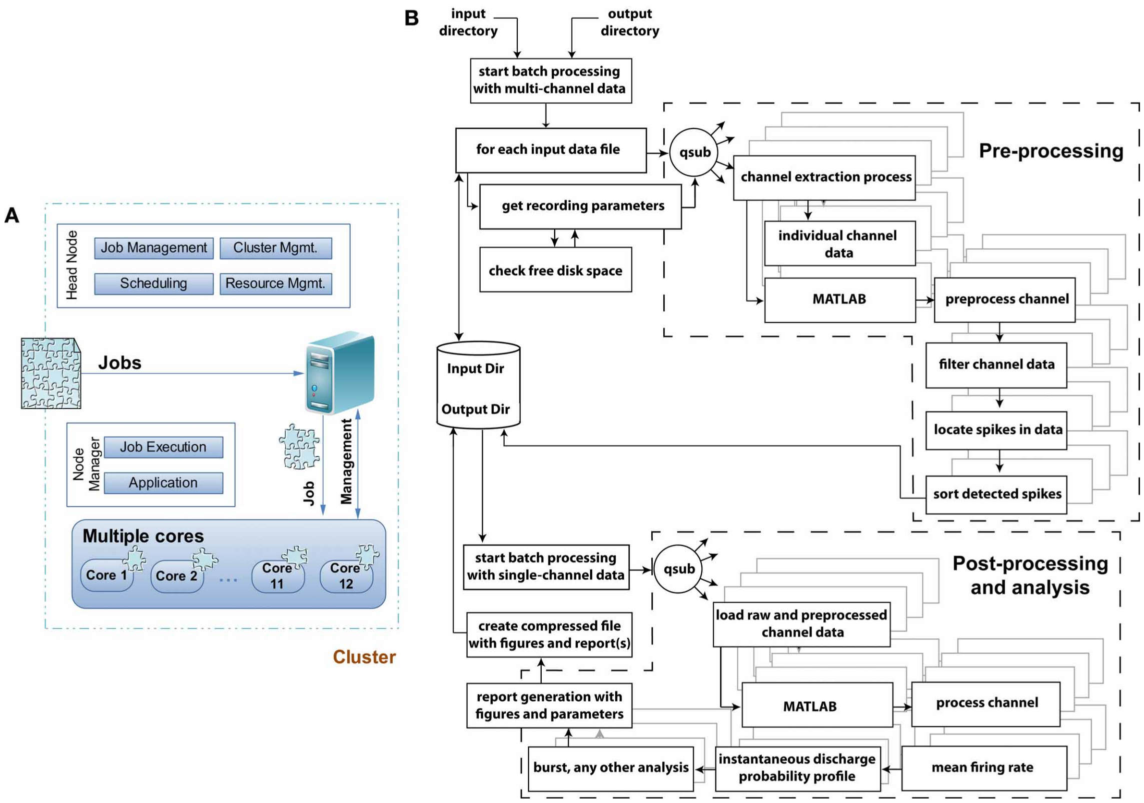


FIGURE 3 | Grid computing architecture and flowchart of the data pre-, post-processing, and analysis module. (A) When jobs arrive at a grid computing environment, the jobs are decomposed into individual jobs and allocated to each processor core or cluster for completion. In a multicomputer cluster system the head node of the cluster is responsible for job-, cluster-, resource-management and scheduling leaving the application and job execution on the individual nodes of the cluster system. In QSpike Tools the benefit of a multicore single computer cluster has been exploited where the responsibilities of the head and individual nodes are handled by the grid computing software. **(B)** Flowchart of the execution model with exemplary pre-, post-processing and analysis modules to demonstrate and distinguish among the operations in terms of their parallelization. Dashed sections outlines the two major sets of operations, where parallelization of the tasks takes place. The remaining tasks instead operate serially. The parallelization is achieved by the *qsub* command, provided by the grid-scheduling environment, distributing the batch-queued jobs to the specified cores for their parallel execution.

redesign of its principle of operation, beyond the embarrassingly parallel computing.

SYSTEM IMPLEMENTATION

The system has been tested on a multi-core personal workstation (Precision, T7500, Dell, Asse-Zellik, Belgium), equipped with two six-core Xeon processors and 24 GBytes of shared memory, running the Ubuntu [5] 10.10 server operating system, the Apache [6] webserver software, and MATLAB R2012a. In addition, a basic grid-computing environment was installed and set up, using the (now outdated) Sun Grid Engine (SGE, Sun Microsystems, Santa Clara, California), or the equivalent Open Grid Scheduler/Grid Engine[7]. The last implements a scheduling system for the management of distributed computing resources (i.e., individual cores, processors, and computers) and it enables the definition of one or more computing queues. Upon launching a job by a special command of the scheduler, while assigning it to a specific queue, the operating system is not anymore in charge of balancing the computing load on the entire computer architecture. Instead, that job is scheduled for execution and assigned to an individual unused node, among those reserved. As mentioned, these nodes may be the processor cores of the server, as in our case, or—transparently for the user—the cores of the processor(s) of Ethernet-connected computers (**Figure 2A**).

Both the client PC(s) and the workstation run the OpenSSH server software, which provides a secure file transfer protocol (SFTP) [8]. With the typical size of a MEA raw data file (e.g., 60 channels/MEA, 16 bits A/D conversion, 20 kHz sampling rate: approximately 8 GB/MEA,h uncompressed), we found that scripted command-line SFTP performed better than drag-and-drop over network mounted shares or graphical user interface clients. Via scripted SFTP and by employing a gigabit Ethernet

[5] http://www.ubuntu.com/download/server

[6] http://httpd.apache.org

[7] http://gridscheduler.sourceforge.net

[8] **(Win)** http://www.cygwin.com, **(OS X)** http://www.maclife.com/article/howtos/how_enable_ssh_your_mac

switch (Catalyst 2960S-48TS-L, Cisco, Diegem, Belgium), the MEA data transfer rate was consistently approximately 100 MBps, in our daily tests.

Figure 3A sketches the simple structure of the grid-computing environment, and of the way we benefit from it. Upon arrival of a preprocessing job request, the large multiplexed (*.mcd) binary file is decomposed into 60 individual files, containing the voltage raw waveforms of each microelectrode. Then, the same operations (**Figure 3B**) are applied and repeated identically to each files, so that the original job is distributed in parallel to the allocated cores. The server performs the management task of submitting the jobs to the execution queue and of checking for a free node (60 channels over, e.g., 12 cores implies an overall load of 5 jobs/core), in a way fully transparent to the end-user. The node manager is ultimately responsible for the execution of each parallelized task, and logs (standard and error) output diagnostics as separate text files.

To favor readability and user customization, most of the data transfers, user communication, job decomposition, and scheduling were coded as Bash shell[9] scripts. Signals preprocessing, analysis, and automated generation of figure plots were implemented as MATLAB scripts.

The flow chart of **Figure 3B** depicts the various components of the preprocessing and analyses pipelines, and indicates (i.e., by dashed line boxes) the execution streams that operate in parallel. As mentioned in the Introduction, the user initiates the execution of a series of sequential steps, where input and output folder names are first provided to the non-interactive Bash scripts, which fetch the (list of) input raw data file(s) and store output results (e.g., the PDF report, the time-stamp of peak-detected multiunit activity, the analog waveform of each detected event for subsequent offline spike-sorting). Then, the individual channels and their preprocessing are distributed simultaneously as independent jobs among the cores using the qsub command, making the analysis trivially parallelized and limited only by the number of nodes available to the queue.

The decomposition of the raw data into the individual channels is the first step in the preprocessing pipeline: it is performed by a custom code, written in C and based on the vendor data access API.

After the extraction of individual channels, further preprocessing like a causal signal filtering (Quian Quiroga, 2009), robust peak-detection detection, elementary spike sorting (i.e., as positive or negative threshold crossings), and spike waveform storage are performed (Quian Quiroga, 2012). For instance, filtering is based on a band-pass, zero-phase digital filter of fourth order (i.e., by filtfilt [10] and ellip [11] MATLAB functions, included in its Signal Processing Toolbox [12]) between 400 and 3000 Hz, while peak detection is based on the evaluation of the median of the raw trace, following the sample code of Wave_clus [13] (Quiroga et al., 2004), which was chosen as our golden standard.

The final result of the peak-detection is the conversion of the analog raw voltage waveforms into a time-series, for each channel: these are the time-stamps of the multiunit activity, which are stored for further analyses as simple text files. As soon as these are available for all channels, they are consolidated in a single file and ready for further MATLAB-based analysis.

The post-processing and analyses pipeline starts with loading the raw single channel data, made available by the channel extraction process, and the files saved during the preprocessing stage. Analyses such as spike-train feature extraction, firing rate estimation, the instantaneous discharge probability profile calculation (Van Pelt et al., 2004), and network-wide synchronized bursting frequency estimation (Bologna et al., 2010) are performed on the preprocessed data. Each of them produces one or more figures, as well as numerical information that are also included in the portable document format (PDF) report generated—see the Supplementary Material. Finally, the report, the figures, and all the intermediate (text and MATLAB binary) files are compressed as a file archive: for each raw data file provided initially, a single file archive is generated by the workflow and available for subsequent user's download.

RESULTS

PERFORMANCES AND SAMPLE PREPROCESSING

The workflow discussed in the previous sections, has been employed daily in our laboratory and extensively tested. For demonstration purposes, we selected typical data files, containing the spontaneous electrical activity of *ex vivo* developing networks of primary cortical neurons, and we provide the PDF report automatically generated by QSpike Tools of its analysis as a Supplementary Information. By a simple web interface, as shown in **Figure 4**, where the various Bash scripts are linked, we experienced that several users with limited computer proficiency could perform smoothly server-side operations, such as visualizing the status of the server queues, clearing the input and output directories from previous data analysis sessions, initiating file transfer, and finally launching the data preprocessing. The same operations could be also performed remotely, form home, upon the virtual private network (VPN[14]) imposed by our university.

Figure 4 shows a screenshots of the web-interface of QSpike Tools. Upon navigating to the web address of the server, the user is presented with page 1, where an identification is required.. This takes the user to the page 2, containing hyperlinks to most of QSpike Tools functionalities: (a) visualizing the current status of the queue, (b) checking the correct availability of the (transferred) files in the input/output directories, (c) transferring data and report to and from the server, (d) clearing the input/output directories of their content upon completion of the analysis, and (e) starting the work-flow by selecting the required number of cores to be used. Once the user opts to transfer raw data files to the server, page 3 appears with additional options. At the end, page 4 is shown with progress and diagnostic information. Finally, as the user wants to download the report, page 5 appears and enables downloading the corresponding file to the user's PC.

[9] http://en.wikipedia.org/wiki/Bash_(Unix_shell)

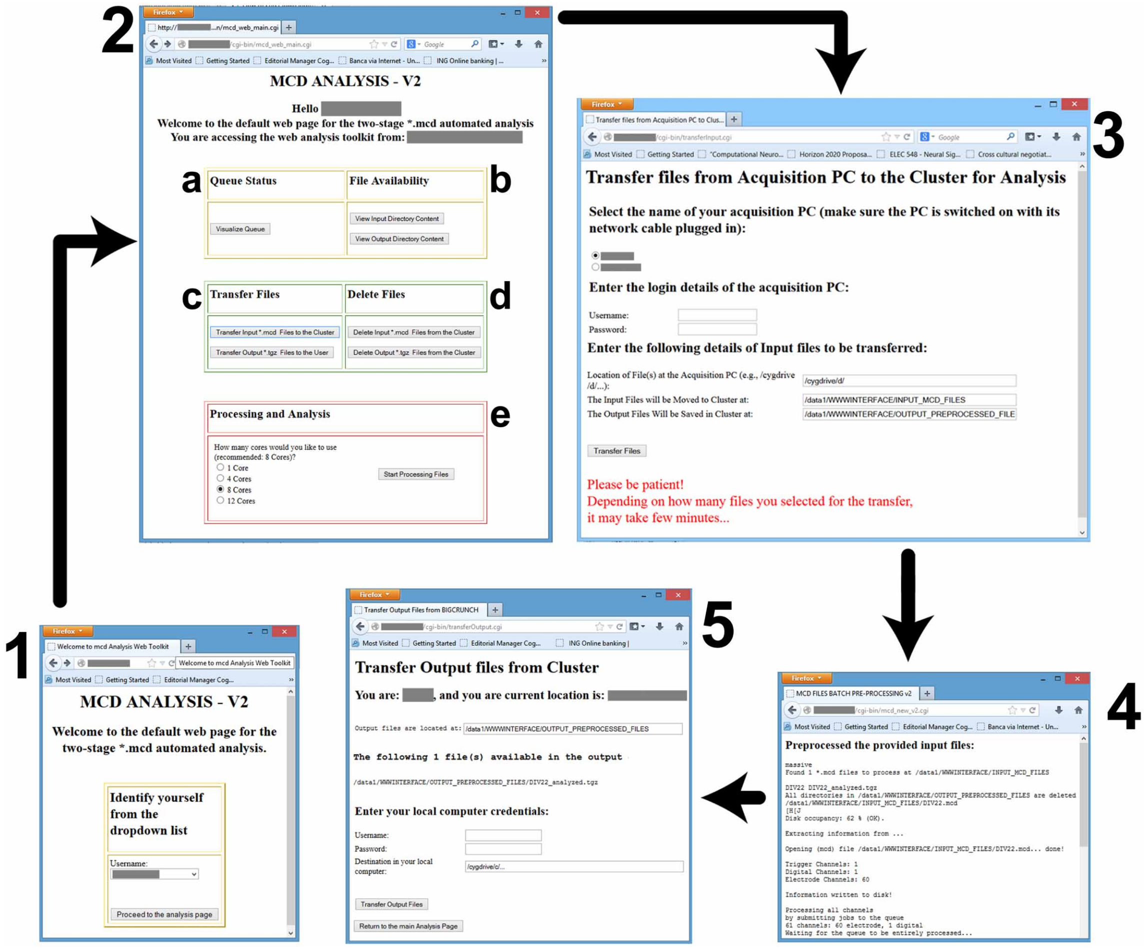
[10] http://www.mathworks.com/help/signal/ref/filtfilt.html

[11] http://www.mathworks.com/help/signal/ref/ellip.html

[12] http://www.mathworks.nl/products/signal/

[13] http://www2.le.ac.uk/centres/csn/wave-clus-docs/

[14] http://en.wikipedia.org/wiki/Virtual_private_network


1
2
3
4
5
a
b
c
d
e
Firefox
Welcome to mcd Analysis Web Toolkit
http://[redacted]...n/mcd_web_main.cgi
[redacted]/cgi-bin/mcd_web_main.cgi
Transfer files from Acquisition PC to Clus...
[redacted]/cgi-bin/transferInput.cgi
MCD FILES BATCH PRE-PROCESSING v2
[redacted]/cgi-bin/mcd_new_v2.cgi
Transfer Output Files from BIGCRUNCH
[redacted]/cgi-bin/transferOutput.cgi
Google
Most Visited
Getting Started
Editorial Manager Cog...
Banca via Internet - Un...
ING Online banking | ...
"Computational Neuro...
Horizon 2020 Proposa...
ELEC 548 - Neural Sig...
Cross cultural negotiat...

1

MCD ANALYSIS - V2

Welcome to the default web page for the two-stage *.mcd automated analysis.

Identify yourself from the dropdown list

Username: [redacted]

Proceed to the analysis page

2

MCD ANALYSIS - V2

Hello [redacted]
Welcome to the default web page for the two-stage *.mcd automated analysis
You are accessing the web analysis toolkit from: [redacted]

a Queue Status: Visualize Queue

b File Availability: View Input Directory Content; View Output Directory Content

c Transfer Files: Transfer Input *.mcd Files to the Cluster; Transfer Output *.tgz Files to the User

d Delete Files: Delete Input *.mcd Files from the Cluster; Delete Output *.tgz Files from the Cluster

e Processing and Analysis

How many cores would you like to use (recommended: 8 Cores)?
○ 1 Core
○ 4 Cores
● 8 Cores
○ 12 Cores

Start Processing Files

3

Transfer files from Acquisition PC to the Cluster for Analysis

Select the name of your acquisition PC (make sure the PC is switched on with its network cable plugged in):

● [redacted]
○ [redacted]

Enter the login details of the acquisition PC:

Username:
Password:

Enter the following details of Input files to be transferred:

Location of File(s) at the Acquisition PC (e.g., /cygdrive /d/...): /cygdrive/d/
The Input Files will be Moved to Cluster at: /data1/WWWINTERFACE/INPUT_MCD_FILES
The Output Files Will be Saved in Cluster at: /data1/WWWINTERFACE/OUTPUT_PREPROCESSED_FILE

Transfer Files

Please be patient!
Depending on how many files you selected for the transfer,
it may take few minutes...

4

Preprocessed the provided input files:

```
massive
Found 1 *.mcd files to process at /data1/WWWINTERFACE/INPUT_MCD_FILES

DIV22 DIV22_analyzed.tgz
All directories in /data1/WWWINTERFACE/OUTPUT_PREPROCESSED_FILES are deleted
/data1/WWWINTERFACE/INPUT_MCD_FILES/DIV22.mcd
[H[J
Disk occupancy: 62 % (OK).

Extracting information from ...

Opening (mcd) file /data1/WWWINTERFACE/INPUT_MCD_FILES/DIV22.mcd... done!

Trigger Channels: 1
Digital Channels: 1
Electrode Channels: 60

Information written to disk!

Processing all channels
by submitting jobs to the queue
61 channels: 60 electrode, 1 digital
Waiting for the queue to be entirely processed...
```

5

Transfer Output files from Cluster

You are: [redacted], and you are current location is: [redacted]

Output files are located at: /data1/WWWINTERFACE/OUTPUT_PREPROCESSED_FILES

The following 1 file(s) available in the output

/data1/WWWINTERFACE/OUTPUT_PREPROCESSED_FILES/DIV22_analyzed.tgz

Enter your local computer credentials:

Username:
Password:
Destination in your local computer: /cygdrive/c/...

Transfer Output Files

Return to the main Analysis Page

FIGURE 4 | Screenshots of the web-interface. Each window is numbered to denote a separate stage of the workflow, and consist in: **(1)** the user-identification; the **(2)** main control webpage; the **(3)** file transfer interface from the data acquisition PC to the master node; **(4)** the result of the preprocessing; and **(5)** the file transfer from the master node to the user PC. Individual letter labels in **(2)** represent grouped functionalities, such as the visualization **(a)** of the status of the computing queues, the availability check **(b)** for the data files in the input and output directories, the file transfer and management functions **(c,d)**, and finally **(e)** the initiation of the parallelized preprocessing and analysis, with an option to select the destination queue.

For the sake of testing and comparison, we configured and run QSpike Tools on sample binary (*.mcd) data files of two different sizes (i.e., approximately 1.5 GB for 8 min and approximately 3.5 GB for 20 min recording). As the execution times depend on both the raw data file and the number of multiunit events, for the sake of fair comparison we executed repeatedly QSpike Tool analysis for 10 times over the very same files. We specifically selected two files, from each file size groups.

We employed four distinct predefined queues (i.e., with 1, 4, 8, and 12 reserved cores) to compare the User Execution Times[15] (**Figure 5**). Confirming the embarrassingly parallelization of the task, we found that execution time reduced significantly ($p < 0.05$, ANOVA; sublinearly) with an increasing number of cores available (see **Figure 5A**), with maximum and minimum execution times ranging from 34.7 min ± 10 s to 10.8 mins ± 17 s or from 31.5 min ± 5 s to 4.3 min ± 6 s for large and small files, respectively.

Despite input files were identical, their repeated analysis led to variable execution times. In order to trace the sources of such variability and provide a preliminary profiler analysis of QSpike Tools, we considered separately the steps performed during the entire workflow, launching manually three subprocesses: (i) raw data channel demultiplexing, (ii) pre-processing of analog voltages (**Figure 3B**), and (iii) post-processing (**Figure 3B**). We then monitored, by standard Linux system calls, the occurrence of three computationally intensive operations managed by the operating systems: voluntary context-switching [16] (VCS), minor page faults [17] (Minor PF), and major page faults (Major PF).

[15] http://en.wikipedia.org/wiki/Time_(Unix)

[16] http://en.wikipedia.org/wiki/Context_switch

[17] http://en.wikipedia.org/wiki/Page_fault

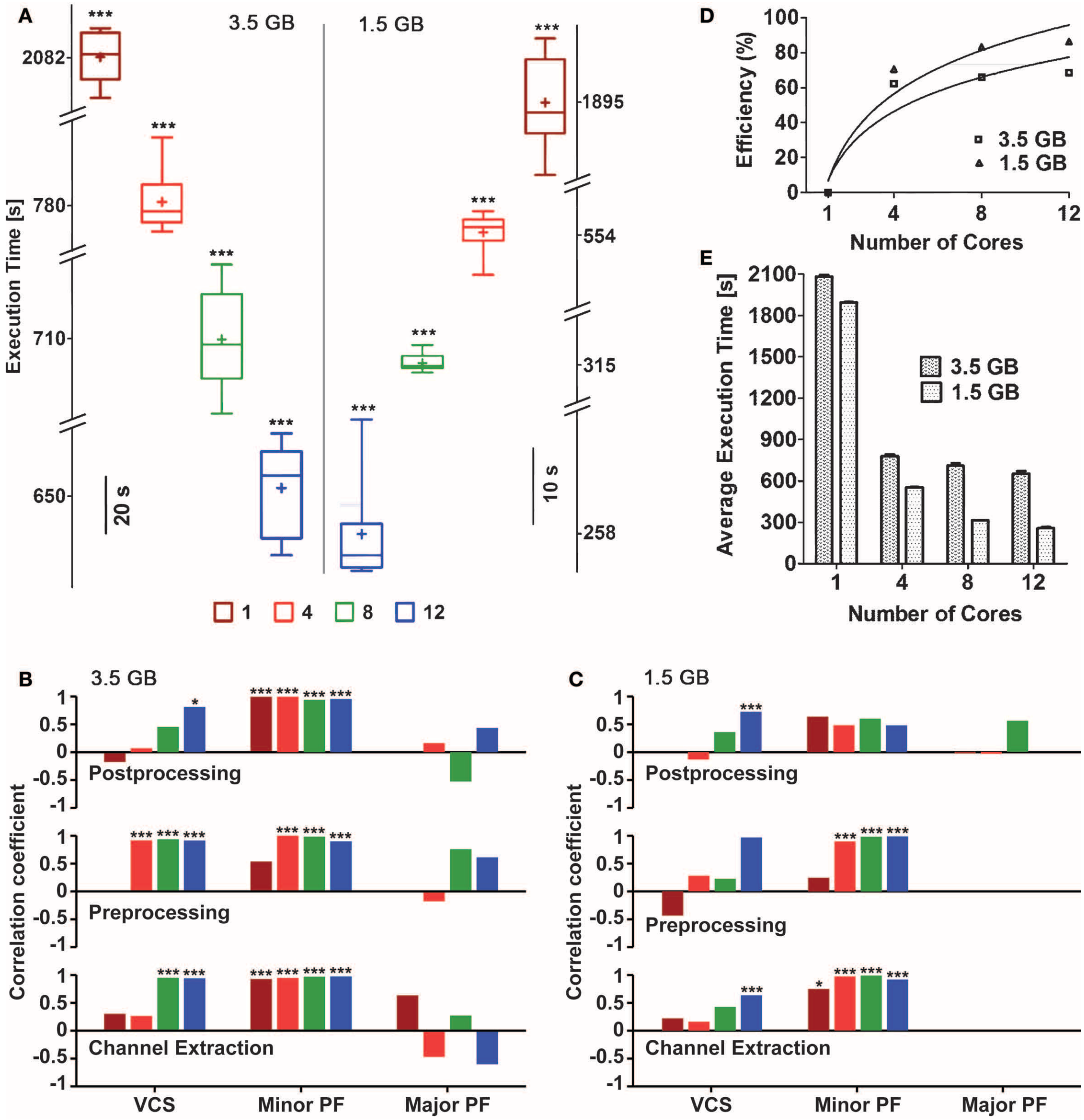


FIGURE 5 | Execution times and efficiency, for an increasing numbers of cores reserved. Box plots **(A)** with box height showing 25–75% of the sample values were used to represent maximum and minimum (whiskers), median ("−") and mean ("+") execution times, respectively, which were all significantly different ($p < 0.05$) based on the number of cores used. The vertical line in the middle separates the data representation corresponding to distinct file sizes (3.5 vs. 1.5 GB). Pearson's liner correlation coefficients **(B,C)** show that the execution times of individual sub-processes are correlated to, certain computationally intensive operations performed by the operating system such as, minor page faults (Minor PF), and voluntary context-switching (VCS) (***$p < 0.0001$; *$p < 0.05$) in case of large file size **(B)**. Though major page faults (Major PF) were noticed while analyzing the large file, they had either negative or no correlation to the execution times. The execution times of the first two sub-processes for the small file were mainly correlated to Minor PF (***$p < 0.0001$; *$p < 0.01$) **(C)**. Overall, negligible amount of Major PF occurred during the execution of the small file and only when large number of cores was used, correlation between execution times and VCS were noticed. The bars in **(B,C)** are plotted using the same color code of **(A)**. The efficiency of parallelization was also quantified **(D)** as referred to the slowest execution time when a single core was used: continuous lines are best-fit logarithmic plots, whose mean squares were 0.8807 and 0.9306 for 3.5 and 1.5 GB file sizes, respectively. The mean execution times **(E)** for both file sizes also show the reduction of the execution time, for an increasing number of cores available.

We found that the execution times are significantly correlated to the occurrence of Minor PF and of VCS, in case of the large input file (**Figure 5B**) ($p < 0.0001$ and $p < 0.05$, respectively). The same occurs for smaller input files, particularly during channel extraction and pre-processing sub-processes (**Figure 5C**).

We then noticed that the execution time with the highest number of cores was found to be more sensitive to page faults and context-switching. This may be explained in terms of the fixed amount of physical memory, as its allocation per core decreases with increasing number of cores. As a consequence, memory

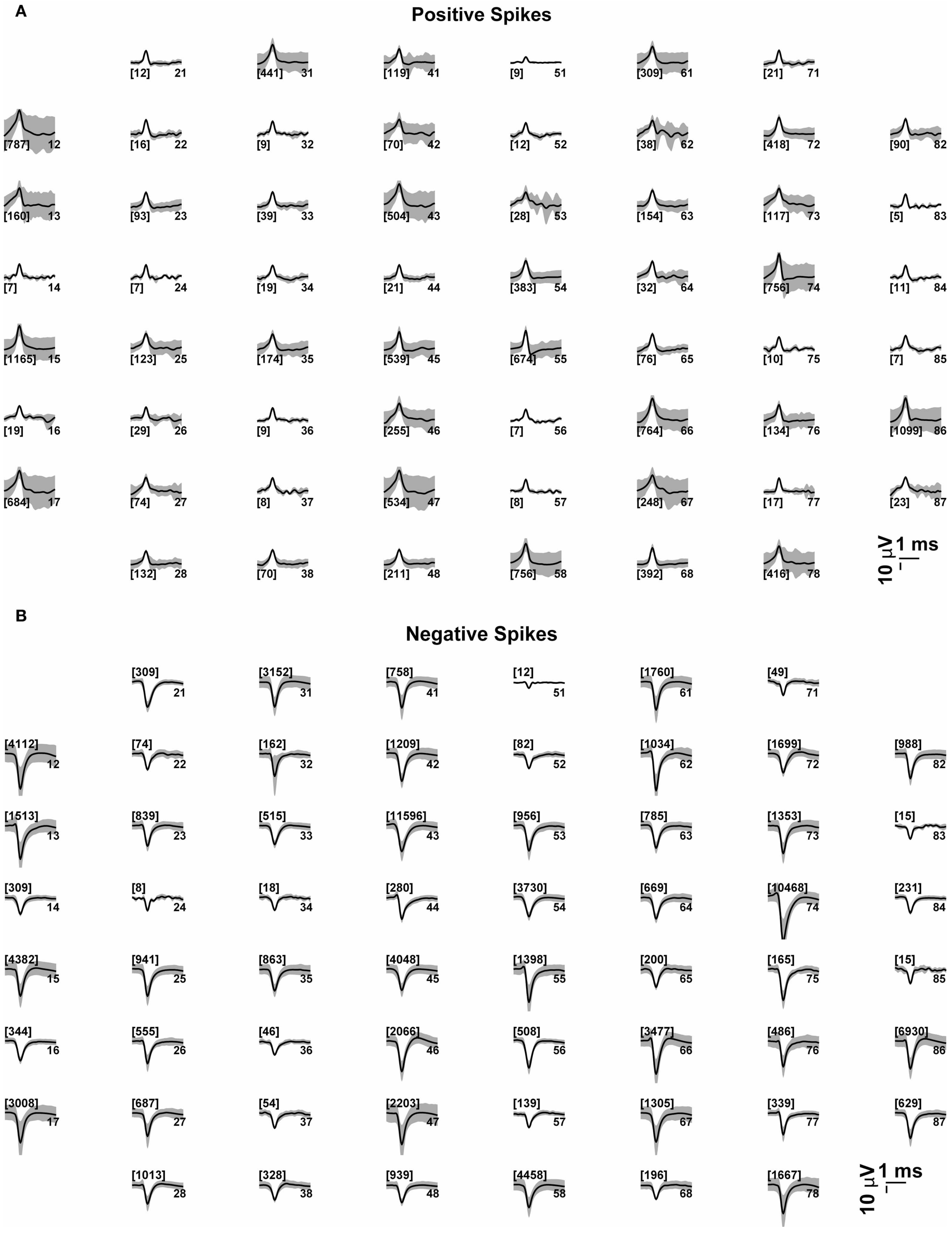


FIGURE 6 | Average voltage waveforms, corresponding to (positive or negative) peak-detection events at each microelectrode. Each subplot represents graphically the 8 × 8 layout of the MEAs employed in this study (see the Methods). The average voltage trajectories (thick black lines) of the positive **(A)** or negative **(B)** threshold crossing events are displayed for each microelectrode, together with its point by point variability (gray shading, i.e., standard deviation), and with the number of events among brackets. The numbers on the right hand side of each subplot denote the labeled electrode name.

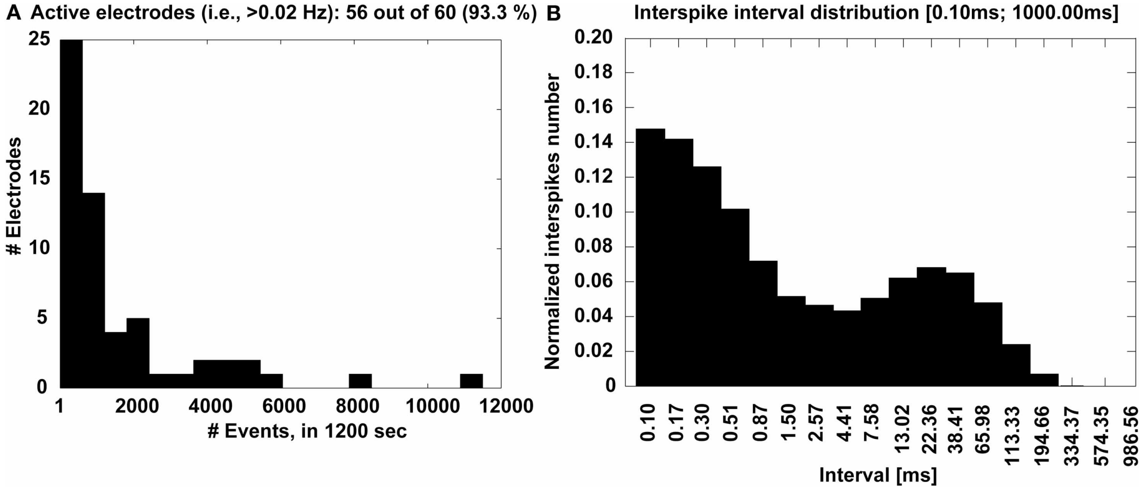


FIGURE 7 | Experimental outcome at a glance. The distribution of the number of electrodes **(A)** that detected activity with significant occurrence frequency is displayed, together with the MEA-wide distribution **(B)** of the interspike intervals that reveal a bimodal profile, corresponding to the recurring transient, MEA-wide synchronization of the electrical activity.

demanding jobs required to run in parallel will have less allotted memory and result in frequent page faults and context-switches (Tay and Zou, 2006). Based on the necessity one may try to reduce the occurrence of page faults by implementing available methods in the literature for better memory management (Zhou et al., 2004).

The efficiency E across distinct queue size, was calculated with respect to the execution times required by a single-core system as $E = T_{max}/\Delta T \times 100\%$, with T_{max} the execution time of the slowest execution—referred to a single core. As expected, the execution efficiency increases (**Figure 5D**) and the mean execution time maintains a decaying trend (**Figure 5E**) with increasing number of cores used.

We have excluded those steps when computing the execution times, such as file transfer to and from the master node file system, since these are dependent on physical characteristics of the Ethernet network as well as of user interactions. The significant differences in the execution times indicate that using more powerful computers with significantly large number of cores is advantageous for a large set of raw data files. We note however that suboptimal memory management by MATLAB parallel instances still deserve attention, as the proportional decrease in the average execution time for large data files differed from the execution time for smaller files.

For the sake of illustration, we further comment and discuss briefly some of the standard analyses performed by QSpike Tools. The first step in the analysis pipeline is to display graphically the waveforms detected at each electrode by the peak-detection algorithm. This allows an elementary spike sorting procedure, discriminating between positive- or negative-threshold crossing. It also enables the user to perform a quantification of the average voltage data trajectory amplitude, its confidence interval, as well as the overall number of events detected. Besides serving as a quality assessment of the raw signals recorded and of the viability of the culture examined (**Figure 6**), next to each subplot the indication of the total number of multiunit events detected at every electrode provides immediate feedback on the significance of the average waveform displayed therein. This is also particularly useful when distinct microelectrode materials are used (e.g., carbon nanotubes or nano-crystalline diamonds coated electrodes), and when the majority of events detected by a given electrodes are monophasic, biphasic, or triphasic extracellular action potentials.

Complementary information is also displayed in the form of a histogram (**Figure 7A**), which quantifies the number of microelectrodes that detected a sufficiently large number of events, higher than a 0.02 Hz minimal occurrence frequency. Finally, the interspike interval (ISI) distribution is also provided across the entire MEAs, merging together all the events (**Figure 7B**), to ultimately reveal whether or not a bimodal distribution is present and corresponds to a mature bursting electrical phenotype of the culture.

Finally, a graphical display of the first few minutes of each recording is also produced, as a raster plot of the individual spike times across channels (**Figure 8**). In our own experience, this provides a rather immediate overview of the recording session and on the viability of the culture. Based on a simple eyeball estimation of the occurrence of MEA-wide synchronized events, and from a detail of the largest synchronized event at increasing temporal resolutions (**Figure 9**), a non-expert user can gain immediate insight on whether the typical expected electrical patterned activity occurred and the extent of the most notable synchronous transient, useful when very small recordings are performed.

The pdf-report, generated automatically at the end of the preprocessing, is provided as Supplementary Information, and offers diagnostic details on the execution of the workflow (i.e., the queue name, start and end time of the processes, and total execution time), as well as of the quantitative information about the data (i.e., total recording duration, total number of samples, sampling

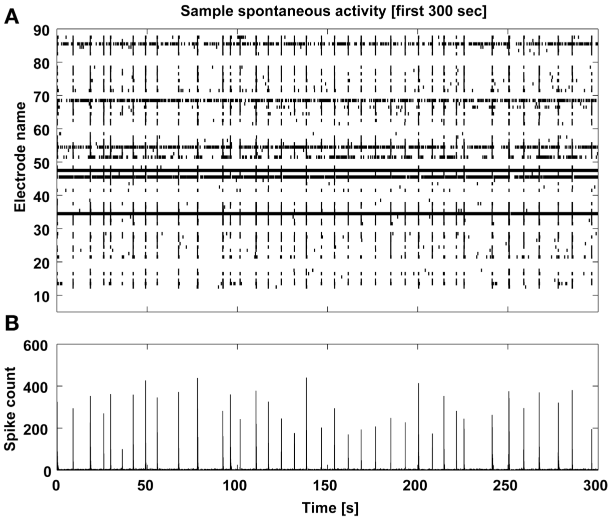


FIGURE 8 | Sample spontaneous activity display. The multiunit spontaneous activity is displayed as raster-plot **(A)** across the detecting microelectrodes for the first 5 min of each data file, and its corresponding spike count is computed **(B)** to reveal the MEA-wise stereotypical episodic synchronization of neuronal activity.

rate, number of active electrodes, number of spikes, number of bursts, mean burst duration, standard deviation of burst duration, mean and standard deviation of the inter-burst-intervals, burst detection threshold, etc.).

DISCUSSION AND CONCLUSION

Starting in the early 2000s, the MEA-Tools (Egert et al., 2002) laid the foundation for user-friendly analysis of MEA data, and eventually became a platform-independent, open source framework for the analysis of neuronal activity data, called FIND (Meier et al., 2008). The toolbox provided for the first time the community with a convenient graphical user interface, and with a set of MATLAB routines, for accessing, visualizing, and analyzing MEAs data. Its minor shortcoming of being centered to a specific hardware system was finally overcome by the DATA-MEAns (Bonomini et al., 2005), which operates and produces ASCII files to be used by other graphical, mathematical or statistical packages. The Neural Signal Manager package (Novellino et al., 2009) was then designed to perform sophisticated analysis of spike and bursting activities, and the SPYCODE package (Bologna et al., 2010) increased the repertoire of standard and advanced tools including cross-correlation analysis and neuronal avalanche detection. MEABench (Wagenaar et al., 2005), on the

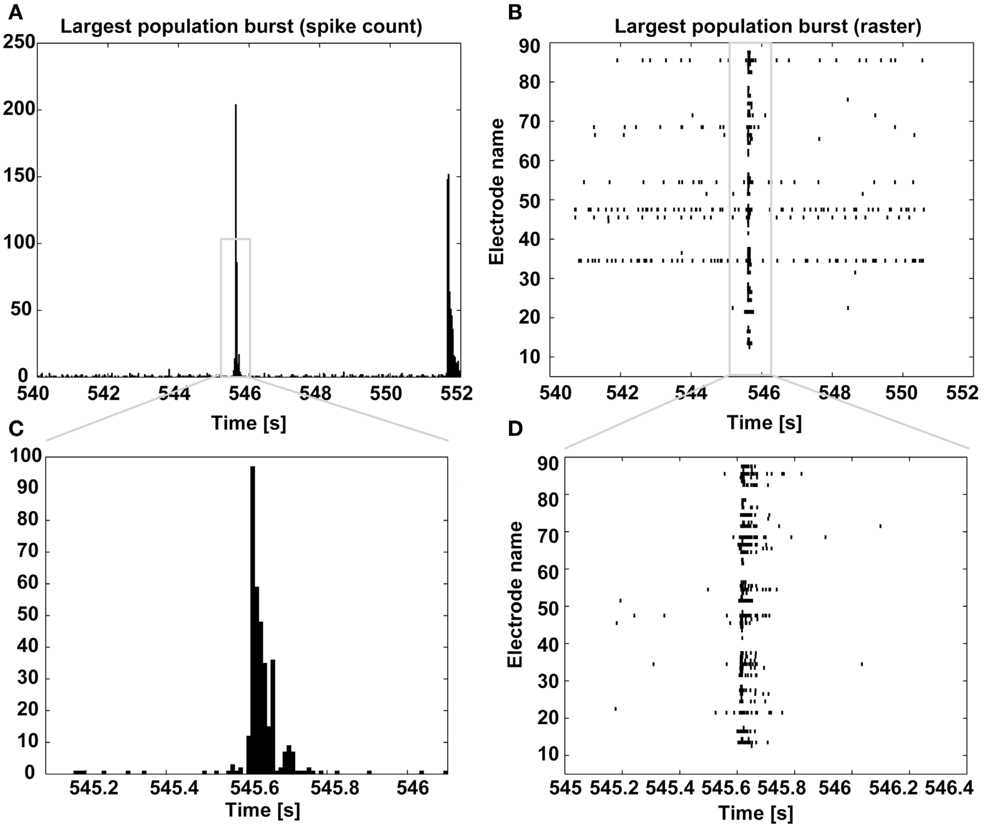


FIGURE 9 | Largest population burst. As in **Figure 8**, the spike count is displayed and it is centered on the largest MEA-wide synchronization event (i.e., a burst) **(A)**, together with its corresponding spike raster diagram **(B)**, with increasing temporal resolution **(C,D)**.

other hand, was designed to provide advanced means at real-time control of the data acquisition hardware, removal of stimulation artifact, detection of spikes, visualization of voltage traces with spike raster plots.

Instead of investing on our own featured package or toolbox, we explicitly focused on the most time-consuming aspect of preprocessing large MEA data files. We do understand that MATLAB is an interpreted language and is not the fastest possible solution to perform stereotyped operations (e.g., filtering and peak-detection). Nonetheless, it has inherent advantages that we strongly favored it as many (new) algorithms are very often proposed, shared, and validated by the community as MATLAB code, making their rapid prototyping or implementation easier. We aimed at taking advantage of those existing analysis tools, but only handling much smaller and portable preprocessed files and we obtained a significant overall reduction in the analysis. For some applications (e.g., a pharma industrial context), however a minimal set of basic analyses and their automated reporting as a PDF file, as we demonstrated here, could instead be sufficient to increase user access and throughput of MEA analysis.

QSpike Tools should be then considered as complementary to existing tools, and its advantages make it suitable for performing automated preprocessing of large datasets, prior to any user interactive (advanced) analysis session.

The limitation of the current version of QSpike Tools is the compatibility with a single proprietary input file format (i.e., *.mcd), as generated by the acquisition software of the commercial platform we use (i.e., Multichannel Systems). Overcoming this limitation is a task for the future and it will be rather simple, in all the cases of raw data formats for which a file interpreter is already available for MATLAB, under Linux. Along these lines, we will also rely on community-supported vendor-neutral initiatives, such as the Neuroshare API library of functions (http://neuroshare.sourceforge.net). In addition, we also aim at (i) enriching the user interface of QSpike Tools, (ii) including a user-specific configuration file to enable custom sets of analyses, and (iii) extending QSpike Tools to *in vivo* data.

Along the lines of the creation of community-supported Neuroinformatics shared facilities for numerical simulations, we launch the proposal for one or more facilities dedicated to automated MEA data analysis. Finally, our work will be made available through the International Neuroinformatics Coordinating Facility, INCF (http://incf.org/) website as well as at https://sites.google.com/site/qspiketool for the community.

INFORMATION SHARING STATEMENT

QSpike Tools and its installation manual are available on request from https://sites.google.com/site/qspiketool.

AUTHORS AND CONTRIBUTORS

This work was carried out in close collaboration between all co-authors. Eleni Vasilaki and Michele Giugliano first defined the research theme and contributed an early software architecture. Mufti Mahmud, Rocco Pulizzi, and Michele Giugliano further implemented and refined methods and algorithms, carried out the data analysis, and preprocessing routines design. Mufti Mahmud and Michele Giugliano wrote the paper. All authors have contributed to, seen and approved the final manuscript.

ACKNOWLEDGMENTS

We are grateful to Dr. G. Musumeci for sharing test electrophysiological recordings, to Mr. D. Van Dyck and Mr. M. Wijnants for excellent technical assistance, and to Dr. L. Gambazzi and Mr. R. A. Barone for prior programming assistance. Financial support from the 7th Framework Programme of the European Commission (Marie Curie Networks "NAMASEN" and "NEUROACT," contracts no. 264872 and 286403; and Future Emerging Technology project "ENLIGHTENMENT" contract no. 284801), the Research Foundation Flanders (grant no. G.0888.12N), and the Interuniversity Attraction Poles Program (IUAP), initiated by the Belgian Science Policy Office, is kindly acknowledged. Access to the supercomputing facilities CalcUA (University of Antwerp) is also acknowledged.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/journal/10.3389/fninf.2014.00026/abstract

REFERENCES

Abdoun, O., Joucla, S., Mazzocco, C., and Yvert, B. (2011). NeuroMap: a spline-based interactive open-source software for spatiotemporal mapping of 2D and 3D MEA data. *Front. Neuroinform.* 4:119. doi: 10.3389/fninf.2010.00119

Arsiero, M., Luscher, H. R., and Giugliano, M. (2007). Real-time closed-loop electrophysiology: towards new frontiers in *in vitro* investigations in the neurosciences. *Arch. Ital. Biol.* 145, 193–209.

Berdondini, L., Chippalone, M., van der Wal, P. D., Imfeld, K., de Rooij, N. F., Koudelka-Hep, M., et al. (2006). A microelectrode array (MEA) integrated with clustering structures for investigating *in vitro* neurodynamics in confined interconnected sub-populations of neurons. *Sens. Actuators B Chem.* 114, 530–541. doi: 10.1016/j.snb.2005.04.042

Berdondini, L., Massobrio, P., Chiappalone, M., Tedesco, M., Imfeld, K., Maccione, A., et al. (2009). Extracellular recordings from locally dense microelectrode arrays coupled to dissociated cortical cultures. *J. Neurosci. Methods* 177, 386–396. doi: 10.1016/j.jneumeth.2008.10.032

Bologna, L. L., Pasquale, V., Garofalo, M., Gandolfo, M., Baljon, P. L., Maccione, A., et al. (2010). Investigating neuronal activity by SPYCODE multi-channel data analyzer. *Neural Netw.* 23, 685–697. doi: 10.1016/j.neunet.2010.05.002

Bonomini, M. P., Ferrandez, J. M., Bolea, J. A., and Fernandez, E. (2005). DATA-MEAns: an open source tool for the classification and management of neural ensemble recordings. *J. Neurosci. Methods* 148, 137–146. doi: 10.1016/j.jneumeth.2005.04.008

Briggman, K. L., and Denk, W. (2006). Towards neural circuit reconstruction with volume electron microscopy techniques. *Curr. Opin. Neurobiol.* 16, 562–570. doi: 10.1016/j.conb.2006.08.010

Buzsaki, G. (2004). Large-scale recording of neuronal ensembles. *Nat. Neurosci.* 7, 446–451. doi: 10.1038/nn1233

Chen, D., Li, X., Cui, D., and Wang, L. (2013a). GPGPU-enabled synchronization measurement of multiple brain regions upon nonlinear interdependence analysis. *IEEE. Trans. Neural Syst. Rehabil. Eng.* 22, 33–43. doi: 10.1109/TNSRE.2013.2258939

Chen, D., Lu, D. C., Tian, M. W., He, S., Wang, S. T., Tian, J., et al. (2013b). Towards energy-efficient parallel analysis of neural signals. *Cluster Comput.* 16, 39–53. doi: 10.1007/s10586-011-0175-6

Chen, D., Wang, L. Z., Ouyang, G. X., and Li, X. L. (2011). Massively parallel neural signal processing on a many-core platform. *Comput. Sci. Eng.* 13, 42–51. doi: 10.1109/MCSE.2011.20

Chen, Y. Y., Lai, H. Y., Lin, S. H., Cho, C. W., Chao, W. H., Liao, C. H., et al. (2009). Design and fabrication of a polyimide-based microelectrode array: application in neural recording and repeatable electrolytic lesion in rat brain. *J. Neurosci. Methods* 182, 6–16. doi: 10.1016/j.jneumeth.2009.05.010

Cui, J., Xu, L., Bressler, S. L., Ding, M., and Liang, H. (2008). BSMART: a Matlab/C toolbox for analysis of multichannel neural time series. *Neural Netw.* 21, 1094–1104. doi: 10.1016/j.neunet.2008.05.007

Denker, M., Wiebelt, B., Fliegner, D., Diesmann, M., and Morrison, A. (2010). "Practically trivial parallel data processing in a neuroscience laboratory," in *Analysis of Parallel Spike Trains,* Vol. 7, eds S. Grün and S. Rotter (New York, NY: Springer US), 413–436. doi: 10.1007/978-1-4419-5675-0_20

Egert, U., Knott, T., Schwarz, C., Nawrot, M., Brandt, A., Rotter, S., et al. (2002). MEA-Tools: an open source toolbox for the analysis of multi-electrode data with MATLAB. *J. Neurosci. Methods* 117, 33–42. doi: 10.1016/S0165-0270(02)00045-6

Fendyur, A., Mazurski, N., Shappir, J., and Spira, M. E. (2011). Formation of essential ultrastructural interface between cultured hippocampal cells and gold mushroom-shaped MEA- towards "IN-CELL" recordings from vertebrate neurons. *Front. Neuroeng.* 4:14. doi: 10.3389/fneng.2011.00014

Fromherz, P. (2006). Three levels of neuroelectronic interfacing. *Ann. N.Y. Acad. Sci.* 1093, 143–160. doi: 10.1196/annals.1382.011

Giugliano, M., La Camera, G., Fusi, S., and Senn, W. (2008). The response of cortical neurons to *in vivo*-like input current: theory and experiment: II. Time-varying and spatially distributed inputs. *Biol. Cybern.* 99, 303–318. doi: 10.1007/s00422-008-0270-9

Gortz, P., Siebler, M., Ihl, R., Henning, U., Luckhaus, C., Supprian, T., et al. (2013). Multielectrode array analysis of cerebrospinal fluid in Alzheimer's disease versus mild cognitive impairment: a potential diagnostic and treatment biomarker. *Biochem. Biophys. Res. Commun.* 434, 293–297. doi: 10.1016/j.bbrc.2013.02.121

Hai, A., and Spira, M. E. (2012). On-chip electroporation, membrane repair dynamics and transient in-cell recordings by arrays of gold mushroom-shaped microelectrodes. *Lab. Chip.* 12, 2865–2873. doi: 10.1039/c2lc40091j

Huang, Y., Li, X., Li, Y., Xu, Q., Lu, Q., and Liu, Q. (2008). An integrative analysis platform for multiple neural spike train data. *J. Neurosci. Methods* 172, 303–311. doi: 10.1016/j.jneumeth.2008.04.026

Hwang, W. J., Lee, W. H., Lin, S. J., and Lai, S. Y. (2013). Efficient architecture for spike sorting in reconfigurable hardware. *Sensors* 13, 14860–14887. doi: 10.3390/s131114860

Jain, T., and Muthuswamy, J. (2008). Microelectrode array (MEA) platform for targeted neuronal transfection and recording. *IEEE Trans. Biomed. Eng.* 55, 827–832. doi: 10.1109/TBME.2007.914403

Just, T., Stoor, S., Klefenz, F., and Husar, P. (2013). Spike sorting algorithm for multichannel MEA recordings based on cross correlation. *Biomed. Tech.* 58(Suppl. 1). doi: 10.1515/bmt-2013-4198

Kim, E. T., Kim, C., Lee, S. W., Seo, J. M., Chung, H., and Kim, S. J. (2009). Feasibility of microelectrode array (MEA) based on silicone-polyimide hybrid for retina prosthesis. *Invest. Ophthalmol. Vis. Sci.* 50, 4337–4341. doi: 10.1167/iovs.08-2500

Kwon, K. Y., Eldawlatly, S., and Oweiss, K. (2012). NeuroQuest: a comprehensive analysis tool for extracellular neural ensemble recordings. *J. Neurosci. Methods* 204, 189–201. doi: 10.1016/j.jneumeth.2011.10.027

Lebedev, M. A., and Nicolelis, M. A. L. (2006). Brain-machine interfaces: past, present and future. *Trends Neurosci.* 29, 536–546. doi: 10.1016/j.tins.2006.07.004

Liu, M.-G., Kang, S.-J., Shi, T.-Y., Koga, K., Zhang, M.-M., Collingridge, G. L., et al. (2013). Long-term potentiation of synaptic transmission in the adult mouse insular cortex: multi-electrode array recordings. *J. Neurophysiol.* 110, 505–521. doi: 10.1152/jn.01104.2012

Magri, C., Whittingstall, K., Singh, V., Logothetis, N. K., and Panzeri, S. (2009). A toolbox for the fast information analysis of multiple-site LFP, EEG and spike train recordings. *BMC Neurosci.* 10:81. doi: 10.1186/1471-2202-10-81

Mahmud, M., Bertoldo, A., Girardi, S., Maschietto, M., and Vassanelli, S. (2012). SigMate: a MATLAB-based automated tool for extracellular neuronal signal processing and analysis. *J. Neurosci. Methods* 207, 97–112. doi: 10.1016/j.jneumeth.2012.03.009

Mazzatenta, A., Giugliano, M., Campidelli, S., Gambazzi, L., Businaro, L., Markram, H., et al. (2007). Interfacing neurons with carbon nanotubes: electrical signal transfer and synaptic stimulation in cultured brain circuits. *J. Neurosci.* 27, 6931–6936. doi: 10.1523/JNEUROSCI.1051-07.2007

Meier, R., Egert, U., Aertsen, A., and Nawrot, M. P. (2008). FIND—a unified framework for neural data analysis. *Neural Netw.* 21, 1085–1093. doi: 10.1016/j.neunet.2008.06.019

Mikula, S., Binding, J., and Denk, W. (2012). Staining and embedding the whole mouse brain for electron microscopy. *Nat. Methods* 9, 1198–1201. doi: 10.1038/nmeth.2213

Morup, M., Hansen, L. K., and Arnfred, S. M. (2007). ERPWAVELAB a toolbox for multi-channel analysis of time-frequency transformed event related potentials. *J. Neurosci. Methods* 161, 361–368. doi: 10.1016/j.jneumeth.2006.11.008

Nicolelis, M. A., and Lebedev, M. A. (2009). Principles of neural ensemble physiology underlying the operation of brain-machine interfaces. *Nat. Rev. Neurosci.* 10, 530–540. doi: 10.1038/nrn2653

Novellino, A., Chiappalone, M., Maccione, A., and Martinoia, S. (2009). Neural Signal Manager: a collection of classical and innovative tools for multi-channel spike train analysis. *Int. J. Adapt. Control Signal Process.* 23, 999–1013. doi: 10.1002/acs.1076

Potter, S., and DeMarse, T. (2001). A new approach to neural cell culture for long-term studies. *J. Neurosci. Methods* 110, 17–24. doi: 10.1016/S0165-0270(01)00412-5

Quian Quiroga, R. (2009). What is the real shape of extracellular spikes? *J. Neurosci. Methods* 177, 194–198. doi: 10.1016/j.jneumeth.2008.09.033

Quian Quiroga, R. (2012). Spike sorting. *Curr. Biol.* 22, R45–R46. doi: 10.1016/j.cub.2011.11.005

Quiroga, R. Q., Nadasdy, Z., and Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput.* 16, 1661–1687. doi: 10.1162/089976604774201631

Rutten, W. L. C. (2002). Selective electrical interfaces with the nervous system. *Annu. Rev. Biomed. Eng.* 4, 407–452. doi: 10.1146/annurev.bioeng.4.020702.153427

Schwartz, A. B. (2004). Cortical neural prosthetics. *Annu. Rev. Neurosci.* 27, 487–507. doi: 10.1146/annurev.neuro.27.070203.144233

Spira, M. E., and Hai, A. (2013). Multi-electrode array technologies for neuroscience and cardiology. *Nat. Nanotechnol.* 8, 83–94. doi: 10.1038/nnano.2012.265

Stevenson, I. H., and Kording, K. P. (2011). How advances in neural recording affect data analysis. *Nat. Neurosci.* 14, 139–142. doi: 10.1038/nn.2731

Tay, Y. C., and Zou, M. (2006). A page fault equation for modeling the effect of memory size. *Perform. Eval.* 63, 99–130. doi: 10.1016/j.peva.2005.01.007

Tian, B. Z., and Lieber, C. M. (2013). Synthetic nanoelectronic probes for biological cells and tissues. *Annu. Rev. Anal. Chem.* 6, 31–51. doi: 10.1146/annurev-anchem-062012-092623

Van Pelt, J., Wolters, P. S., Corner, M. A., Rutten, W. L. C., and Ramakers, G. J. A. (2004). Long-term characterization of firing dynamics of spontaneous bursts in cultured neural networks. *IEEE Trans. Biomed. Eng.* 51, 2051–2062. doi: 10.1109/TBME.2004.827936

Vassanelli, S., Mahmud, M., Girardi, S., and Maschietto, M. (2012). On the way to large-scale and high-resolution brain-chip interfacing. *Cogn. Comput.* 4, 71–81. doi: 10.1007/s12559-011-9121-4

Vato, A., Bonzano, L., Chiappalone, M., Cicero, S., Morabito, F., Novellino, A., et al. (2004). Spike manager: a new tool for spontaneous and evoked neuronal networks activity characterization. *Comput. Neurosci. Trends Res.* 1153–1161. doi: 10.1016/j.neucom.2004.01.180

Wagenaar, D., DeMarse, T. B., and Potter, S. M. (2005). "MeaBench: a toolset for multi-electrode data acquisition and on-line analysis," in *2nd Internatinoal Ieee/Embs Conference on Neural Engineering*, (Arlington, TX), 518–521. doi: 10.1109/CNE.2005.1419673

Wang, J., Wagner, F., Borton, D. A., Zhang, J., Ozden, I., Burwell, R. D., et al. (2012). Integrated device for combined optical neuromodulation and electrical recording for chronic *in vivo* applications. *J. Neural Eng.* 9:016001. doi: 10.1088/1741-2560/9/1/016001

Wilson, J. A., and Williams, J. C. (2009). Massively parallel signal processing using the graphics processing unit for real-time brain-computer interface feature extraction. *Front. Neuroeng.* 2:11. doi: 10.3389/neuro.16.011.2009

Wise, K. D., Anderson, D. J., Hetke, J. F., Kipke, D. R., and Najafi, K. (2004). Wireless implantable microsystems: high-density electronic interfaces to the nervous system. *Proc. IEEE* 92, 76–97. doi: 10.1109/JPROC.2003.820544

Yu, B., Mak, T., Li, X., Smith, L., Sun, Y., and Poon, C. S. (2012). Stream-based Hebbian eigenfilter for real-time neuronal spike discrimination. *Biomed. Eng. Online* 11:18. doi: 10.1186/1475-925X-11-18

Zhou, P., Pandey, V., Sundaresan, J., Raghuraman, A., Zhou, Y., and Kumar, S. (2004). "Dynamic tracking of page miss ratio curve for memory management," in *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems* (Boston, MA: ACM), 177–188.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 12 December 2013; accepted: 27 February 2014; published online: 19 March 2014.

Citation: Mahmud M, Pulizzi R, Vasilaki E and Giugliano M (2014) QSpike tools: a generic framework for parallel batch preprocessing of extracellular neuronal signals recorded by substrate microelectrode arrays. Front. Neuroinform. ***8****:26. doi: 10.3389/fninf.2014.00026*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

REVIEW ARTICLE
published: 26 March 2014
doi: 10.3389/fninf.2014.00029

A review of multivariate analyses in imaging genetics

Jingyu Liu[1,2] * *and Vince D. Calhoun*[1,2]

[1] *The Mind Research Network and Lovelace Biomedical and Environmental Research Institute, Albuquerque, NM, USA*
[2] *Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA*

Edited by:
Venkata Satyanand Mattay, Lieber Institute for Brain Development, USA

Reviewed by:
Yi-Yuan Tang, Institute of Neuroinformatics, Dalian University of Technology, China
Fabio Sambataro, Istituto Italiano di Tecnologia, Italy
Qiang Chen, Lieber Institute for Brain Development, USA

****Correspondence:***
Jingyu Liu, Mind Research Network and Lovelace Biomedical and Environmental Research Institute, 1101 Yale Boulevard NE, Albuquerque, NM 87106, USA
e-mail: jliu@mrn.org

Recent advances in neuroimaging technology and molecular genetics provide the unique opportunity to investigate genetic influence on the variation of brain attributes. Since the year 2000, when the initial publication on brain imaging and genetics was released, imaging genetics has been a rapidly growing research approach with increasing publications every year. Several reviews have been offered to the research community focusing on various study designs. In addition to study design, analytic tools and their proper implementation are also critical to the success of a study. In this review, we survey recent publications using data from neuroimaging and genetics, focusing on methods capturing multivariate effects accommodating the large number of variables from both imaging data and genetic data. We group the analyses of genetic or genomic data into either *a priori* driven or data driven approach, including gene-set enrichment analysis, multifactor dimensionality reduction, principal component analysis, independent component analysis (ICA), and clustering. For the analyses of imaging data, ICA and extensions of ICA are the most widely used multivariate methods. Given detailed reviews of multivariate analyses of imaging data available elsewhere, we provide a brief summary here that includes a recently proposed method known as independent vector analysis. Finally, we review methods focused on bridging the imaging and genetic data by establishing multivariate and multiple genotype-phenotype-associations, including sparse partial least squares, sparse canonical correlation analysis, sparse reduced rank regression and parallel ICA. These methods are designed to extract latent variables from both genetic and imaging data, which become new genotypes and phenotypes, and the links between the new genotype-phenotype pairs are maximized using different cost functions. The relationship between these methods along with their assumptions, advantages, and limitations are discussed.

Keywords: imaging genetics, multivariate analyses, genotype, phenotype, intermediate phenotypes

INTRODUCTION

While most genetic studies have focused on phenotypes as diagnoses and clinical symptoms, it is relatively recent that intermediate phenotypes have become an ever increasing focus. Intermediate phenotypes refer to biological trait phenotypes conveying relatively closer association or higher penetration than traditional phenotypes (Meyer-Lindenberg and Weinberger, 2006; Rasetti and Weinberger, 2011). The best examples of approaches leveraging intermediate phenotypes come from studies of psychiatric disorders for which diagnoses are based mainly on clinical observations and interviews. Intermediate phenotypes derived from neuroimaging and signals directly assessing brain structure and function not only reduce the phenotypic heterogeneity common to many psychiatric disorders, but also increase detection power, given the genetic effects are not expressed directly as behaviors but as molecular and cellular functions mediating brain development and processes (Gottesman and Gould, 2003; Rose and Donohoe, 2013). The pioneer studies utilizing neuroimaging features to identify genetic impact were in the year 2000 (Bookheimer et al., 2000; Heinz et al., 2000; Small et al., 2000). They signified the birth of a new research approach using imaging genetics. As defined (Hariri et al., 2006; Meyer-Lindenberg et al., 2008; Silver et al., 2011; Meyer-Lindenberg, 2012), it combines genetic information and neuroimaging data in the same subjects to discover neuromechanisms linked to psychiatric disorders. The overall strength of imaging genetics and its impact on psychiatric disorder studies or broader have been stated clearly in several reviews (Meyer-Lindenberg and Weinberger, 2006; Glahn et al., 2007; Bigos and Weinberger, 2010; Meyer-Lindenberg, 2010; Rasetti and Weinberger, 2011).

The overwhelming growth of imaging genetics in recent years as summarized in recent studies (Roffman et al., 2006; Bigos and Weinberger, 2010), while providing abundant promising results, also reveals challenges embedded within study designs such as validity of candidate genes, control of non-genetic confounding factors, and selection of tasks to stimulate brain specific processes. Bigos and Weinberger (2010) have provided an excellent review with applications to demonstrate the principles in designing an imaging genetic study. Another big challenge faced by both imagers and geneticists is how to properly analyze the collected data, since both neuroimaging and genetics tend to generate a large amount of data. Different strategies, processing approaches, and validation methods such as false positive control (Silver et al., 2011) have been implemented and tailored for different conditions. But there is an even greater need in the future for the methodology development as pointed by Mayer-Lindenberg in his recent review (Meyer-Lindenberg, 2012), where complexity of epistasis, pleiotropy and genetic by environment interactions should been considered in particular in large scale genomic studies. The availability of imaging genetic analytic tools and their proper implementation are critical for both success of individual studies and the continuing growth of imaging genetics.

The earliest imaging genetic studies focused on candidate genetic variants using either a single or a few variables (Bookheimer et al., 2000; Heinz et al., 2000; Small et al., 2000; Egan et al., 2001). For example, the dopamine transporter gene (SLC6A) was analyzed with neuroimaging data from single-photon emission computed tomography (Heinz et al., 2000). Variation within the APOE gene was associated with activities in memory function affected by Alzheimer's disease (Bookheimer et al., 2000). COMT Val allele carriers showed increased activities in the prefrontal cortex compared to Met allele carriers (Egan et al., 2001). In parallel, the intermediate phenotypes from neuroimaging techniques can also be specified within selected brain regions or particular processes. Straightforward univariate analyses are often used and well suited for these studies. Candidate gene and candidate imaging phenotype studies in the last decade have proven the validity of imaging genetic approach as recapitulated in (Meyer-Lindenberg, 2012). But with the completion of human genome sequence and multimodal imaging practices, in conjunction with increased evidence of polygenicity and pleiotropy (Purcell et al., 2009; Sivakumaran et al., 2011; Whalley et al., 2012; Smoller et al., 2013), multivariate analysis methods are becoming more and more demanding. For instance, thousands of genetic variants have been suggested to be linked with the risk for schizophrenia (Purcell et al., 2009). Methods to capture the interactive or integrated genetic effects of a set of genetic variants, methods to extract brain networks formed from individual voxels or regions, and methods to detect, possibly, multiple genotype-phenotype connections have been developed with their limitations and advantages (Hardoon et al., 2009; Liu et al., 2009b; Vounou et al., 2010; Le Floch et al., 2012). We expect to see continued development of such powerful methods to face the challenges and promises from genome-wide whole brain association studies.

In this review, we focus on analysis approaches and, more specifically, on the multivariate analysis approaches. We will first give an overview of analysis strategies. Then, we will survey the methods and organize them according to their multivariate nature on genetic data, neuroimaging data or both.

OVERVIEW OF ANALYSIS STRATEGIES IN IMAGING GENETICS

While various strategies can be applied to design and perform imaging genetic studies, several aspects of such studies require particular caution. Firstly, when an imaging feature is selected as the intermediate or endophenotype, useful criteria should be applied or at least considered. As summarized in (Gottesman and Gould, 2003) intermediate phenotypes should show association with illness in a population, certain level of heritability, and state-independent characters. A proper preprocessing or controlling for possible confounding factor should also be in place, such as scanning effects, age or gender difference, brain size, etc. The most often used software packages to process brain imaging data, particularly for magnetic resonance imaging (MRI) images, include FSL[1] , SPM[2], and AFNI[3] for functional and structural voxel-wise preprocessing, and FreeSurfer[4] for brain regional volume and cortical thickness. Secondly, genetic data either from single genetic mutation or genomic variants should be checked for family structure, population structure, and ethnicity differences. A rationale to pull samples together should be justified through, for instance, from a homogenous group, no indication of population structure, or a proper control of ethnicity difference. The most often used software package for single nucleotide polymorphism (SNP) data is plink[5], which provides tools to do various quality control, sample relatedness tests, filtering and population stratification. The most often used software packages (freely available) for calling copy number variation (CNV) include PennCNV[6], and BirdSuite[7]. Even though the effect of CNVs on brain imaging phenotypes is understudied now, it has been predicted to be an important extension in the future (Meyer-Lindenberg, 2012). Thirdly, methods to test the relation between genetics and imaging phenotypes heavily rely on the dimensionality of data, as explained explicitly in next paragraph. Finally, the interpretation of results depends on the study design and analysis approaches. Keep in mind that most imaging genetic studies test the association between genetic variants and imaging phenotypes, as the analytical method itself reveals later on. Any causal relation and underlying biological mechanism is only suggestive. Particular caution should be given to genome-wise association studies which result in a set of genetic variants interactively associated with imaging phenotypes. The interaction among them, linear, non-linear, dominate, recessive, two-way or n-way, etc., needs to be carefully explained and some methods test the overall effect without knowing the detailed interrelations. The verification or at least certain levels of cross evaluation for such findings as described in (Le Floch et al., 2012) plays a very crucial role.

Depending on the dimensionality of investigated genotypes and imaging intermediate phenotypes, we can classify imaging genetic studies into four categories, which is a concept borrowed from Vounou et al. (2010). As plotted in **Figure 1**, the first one includes studies with candidate phenotypes and candidate genotypes, where a direct univariate association test is applied to assess the hypothesized connection. A control for possible confounding factors (scanner, age, gender, medication, etc.) should be considered for imaging phenotypes. The second type includes studies investigating multiple genetic variants, ranging from a few to 100s of 1000s of variables in a genome-wide setting. Univariate tests corrected for multiple comparisons are straightforward (Potkin et al., 2009), but it may miss the well

[1] http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/

[2] http://www.fil.ion.ucl.ac.uk/spm/

[3] http://afni.nimh.nih.gov/afni/

[4] https://surfer.nmr.mgh.harvard.edu/

[5] http://pngu.mgh.harvard.edu/∼purcell/plink/

[6] http://www.openbioinformatics.org/penncnv/

[7] http://www.broadinstitute.org/scientific-community/science/programs/medical-and-population-genetics/birdsuite/birdsuite

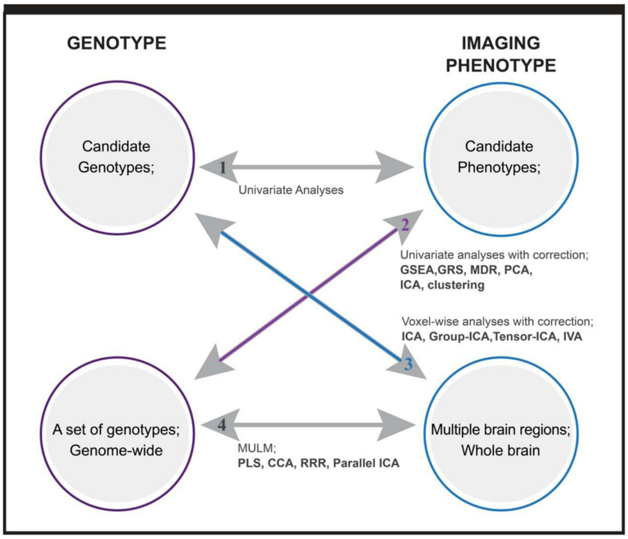

GENOTYPE
IMAGING PHENOTYPE
Candidate Genotypes;
Candidate Phenotypes;
1
Univariate Analyses
2
Univariate analyses with correction;
GSEA,GRS, MDR, PCA, ICA, clustering
Voxel-wise analyses with correction;
ICA, Group-ICA,Tensor-ICA, IVA
3
A set of genotypes; Genome-wide
4
MULM;
PLS, CCA, RRR, Parallel ICA
Multiple brain regions; Whole brain

FIGURE 1 | Overview of imaging genetic studies and methods applied. Category 1: candidate genotype with candidate phenotype. Category 2: sets of genotypes with candidate phenotype. Category 3: candidate genotype with multiple imaging phenotypes. Category 4: sets of genotypes with multiple imaging phenotypes. Methods written in bold are multivariate analysis methods. GSEA: gene set enrichment analysis; GRS: genetic risk score; MDR: multifactor dimensionality reduction; PCA: principal component analysis; ICA: independent component analysis; IVA: independent vector analysis; MULM: mass univariate linear model; PLS: partial least square; CCA: canonical component analysis; RRR: reduced rank regression.

documented gene–gene interactions. Data driven multivariate methods and *a priori* based gene-set or pathway analyses are the two main analytical approaches to capture the interactive or integrated genetic effect (Liu et al., 2010b; Walton et al., 2013). What type of interactive relation among genes can be captured depends on the analytic methods or specifically, the models that the methods are built on. The third type includes studies investigating multiple imaging phenotypes, which may come from one or more imaging modalities, such as structural, functional MRI, magnetic resonance spectroscopy, etc. The imaging phenotypes may cover whole brain or many brain regions or voxels. Except for voxel-wise analyses with multiple comparison correction, the strategy to analyze such phenotypes usually is to extract brain networks formed by interactive brain regions or voxels, thus not only accommodating interrelations but also reducing the number of tested phenotypes (Calhoun and Adali, 2006). The last group of studies involves associations between multiple genotypic variables and multiple phenotypic variables. A typical example is genome-wide whole brain studies. Although massive univariate approaches have been implemented such as a mass-univariate linear model (MULM) in studies (Stein et al., 2010), most utilize data reduction and factorization methods to effectively capture the interactive and complex relations within and between datasets. In the following, we present the analytical methods implemented in studies of the last three categories, category 2: sets of genotypes with candidate phenotype, category 3: candidate genotype with multiple imaging phenotypes, and category 4: sets of genotypes with multiple imaging phenotypes. We focus on the multivariate approaches for each category.

A priori BASED MULTIVARIATE ANALYSES ON GENETIC/GENOMIC DATA (CATEGORY 2)

Gene set enrichment analysis (GSEA) is a computational method that determines whether a prior defined set of genetic variants shows statistically significant differences between two biological states (Mootha et al., 2003; Subramanian et al., 2005) or, more generally, significant associations with phenotypes compared to the null hypothesis. The GSEA was first introduced in cancer research and thereafter various modified versions have been introduced in studies of different diseases that includes psychiatric disorders (Subramanian et al., 2005; Holden et al., 2008; Suarez-Farinas et al., 2010; Oh et al., 2011; Weng et al., 2011). The basic principle of GSEA is that sets of genetic variants are first selected for tests. We will use SNPs as an example of genetic variants without loss of generality in this review. A set of SNPs are selected based on common biological attributes (gene ontology or pathways), chromosome location, or reported results in the literature. Then the overrepresentation, or "enrichment," of phenotype-association of this set of SNPs as one unit is calculated against the null hypothesis of normally distributed phenotype-association. Among many ways to decide the significance of enrichment (Abatangelo et al., 2009), the two most common methods are Fisher's exact test and enrichment score test (Subramanian et al., 2005). Fisher's exact test is fast but needs a pre-defined threshold, while enrichment score does not need a threshold but needs a permutation to get empirical p values. Specific issues associated, such as gene size bias (Mirina et al., 2012), linkage disequilibrium (LD) between adjacent SNPs, have been addressed by various modified versions (Liu et al., 2010b; Li et al., 2011). The rationale to select the set of SNPs comes from prior information, so this approach is indeed *a priori* driven test for the overall effect of multiple variables, without modeling the exact interaction among them. Another similar approach proposed by Walton et al. (2013) is to compute a cumulative genetic risk score ($GRS = \sum_{i=1}^{N} w^i x^i$), which combines the additive effects of multiple SNPs selected from the continuously updated meta-analysis of genetic studies. The authors showed that this multivariate score combined the impact of many genes with small effects, accounting for 3.6% of the total variance of brain activity at dorsal lateral prefrontal cortex (Walton et al., 2013). Similar approaches using polygenic risk scores have been implemented in several other studies (Whalley et al., 2012; Smoller et al., 2013).

DATA DRIVEN MULTIVARIATE ANALYSES ON GENETIC/GENOMIC DATA (CATEGORY 2)

Unlike the approaches above, some studies have implemented purely data driven analyses without prior information, emphasizing the genetic patterns embedded in the datasets to capture the epistasis and polygenicity. Multifactor dimensionality reduction (MDR) was developed to identify combinations of gene–gene and gene-environmental factors that are predictive of a phenotype (Hu et al., 2011; Gui et al., 2013; Pan et al., 2013). The heart of MDR is an attribute construction algorithm that creates a new variable by pooling genotypes from multiple SNPs (Moore et al., 2010). In brief, values from any combination of multiple SNPs are classified into two distinct groups, high risk and low risk, effectively reducing the dimensionality from multidimensional to one-dimensional. Subsequently, the new variables are used to identify, from all potential combinations, the specific combination of SNPs showing the strongest association with the phenotype. This method with no particular model assumption is well suited for capturing epistasis and has been used in genetics studies of various disease status (Ritchie et al., 2001; Moore and Williams, 2002; Ma et al., 2005; Lou et al., 2007; Gui et al., 2011). Extensions of the method have been developed for quantitative phenotypes and genome-wide data (Lou et al., 2007; Pattin et al., 2009; Cattaert et al., 2011; Oh et al., 2012; Winham, 2013). It is expected to see more broad applications of this method even in imaging genetics (Papassotiropoulos and de Quervain, 2011). Within the same line of estimating aggregated effect of multiple genetic variants, but based on a linear additive model, multiple regression and its penalized or modified versions have been implemented to assess the explanation power of gene variables (from a couple to genome-wide) to various of phenotypes (Wang and Abbott, 2008; Wu et al., 2009; Cule et al., 2011). Penalized regression, specifically LASSO multiple regression, are also often used to downsize variables (voxels or SNP) for further analyses (Vounou et al., 2012).

Other types of data-driven approaches, as reviewed in (Jombart et al., 2009), mainly include principal component analysis (PCA), principal coordinate analysis, non-metric dimensional scaling, and correspondence analysis, belonging to the category of matrix decomposition and extracting factors/components of weighted genetic variants. An addition to the review is independent component analysis (ICA). PCA provides a set of linearly orthogonal principal components, explaining maximal variance, while ICA is designed to extract statistically independent components (and thus uses higher order statistical information). PCA is often used in genome-wide SNP data, and the top PCs extracted most likely present the population structure helpful for population stratification (Price et al., 2006; Liu et al., 2010a). ICA has proven successful in a variety of biological inquiries when applied to gene expression data (Kong et al., 2008), including identifying tumor-related pathways (Saidi et al., 2004; Sheng et al., 2011), classifying disease datasets (Huang and Zheng, 2006) and mining human gene expression modules (Engreitz et al., 2010).

The value of clustering methods has been established in various genetic studies, as reviewed by Jiang et al. (2004), as a means to group genetic variants according to their functional relatedness (D'haeseleer, 2005). In an example of using imaging as phenotypes, Sloan et al. (2010) applied a hierarchical clustering analysis on 834 SNPs and clinical and imaging phenotypes, including left, right hippocampal volume and gray matter density. The association between each SNP and each endpoint was first computed, and then the clustering was performed on the results, wherein both genotypes and phenotypes were grouped based on similarity. Subsequently, p-values for each cluster were estimated using bootstrap resampling. This study showed that (1) SNPs are frequently associated with imaging phenotypes and rarely associated with clinical scores and (2) most of the genes found within clusters are associated with either beta-amyloid production or apoptosis (Sloan et al., 2010). A noteworthy point of this study is that it combined a pathway-based approach and clustering analyses together, first by selecting SNPs based on pathways and then applying clustering on genotypes and phenotypes, and demonstrated that priori driven and data driven approaches can be integrated into one study.

COMPONENT-BASED ANALYSES ON IMAGING DATA (CATEGORY 3)

Not only does the development of various neuroimaging techniques improve the precision of measurement of brain attributes, but it also stimulates the growth of analysis approaches. The most common imaging modalities include functional MRI (fMRI), measuring the dynamic brain activity based on blood-oxygenation-level dependent contrast; structural MRI, assessing the volume and density of gray matter, white matter, and cerebrospinal fluid; diffusion (tensor) imaging, depicting the white matter tract connections; and magnetic resonance spectroscopy, obtaining biochemical information about the tissues of brain. Furthermore, collecting multiple types of imaging data from the same individuals becomes a common practice in the hope of revealing additional information and increasing our knowledge. Thus, methods for multimodal analyses have also emerged and developed rapidly. Here, we limit ourselves to the component-based multivariate analysis approaches applied to imaging data, though there are many other multivariate approaches, such as unsupervised clustering, supervised pattern recognition, classification and projection, and others (Dimitriadou et al., 2004; Demirci et al., 2008; Hinrichs et al., 2009; Filipovych and Davatzikos, 2011).

ICA with various implementation algorithms (Cardoso, 1997; Hyvirinen and Oja, 1999; Bingham and Hyvarinen, 2000) and its modifications and extensions (Bach and Michael, 2002; Beckmann and Smith, 2004; Calhoun et al., 2005; Hong et al., 2005; Lin et al., 2010) are the most popular methods for multivariate analyses on imaging data. Several reviews have been offered to the imaging field (McKeown et al., 2003; Calhoun and Adali, 2006; Calhoun et al., 2009). Here, we briefly summarize the main points. A typical ICA model assumes that the source signals are not observable, statistically independent and non-Gaussian with an unknown but linear mixing process. Consider an observed M–dimensional random vector denoted by $X = [x_1, x_2,...,x_M]^T$, which is generated by the ICA model: $X = AS$, S is the source matrix. The goal of ICA is to estimate an unmixing matrix W such that Y given by $Y = WX$ is a good approximation to the "true" sources. Y is called the component matrix. In the context of imaging data, components are the independent brain networks embedded in the observed voxels. Furthermore, when MRI data from multiple subjects, each with their own temporal dynamics, are of interest, several ICA based multi-subject analysis approaches have been proposed (Calhoun et al., 2001; Schmithorst and Holland, 2004; Beckmann and Smith, 2005; Esposito et al., 2005; Erhardt et al., 2011; Calhoun and Adali, 2012). We refer to recent studies by Calhoun and Adali (2012); (Calhoun et al., 2009) for a more detailed explanation. A recent addition is independent vector analysis (IVA), which is a generalization of ICA for analysis of multiple datasets (Kim et al., 2006). It takes a model of $X^{[m]} = A^{[m]}S^{[m]}$, $Y^{[m]} = W^{[m]}X^{[m]}$, where M is the number of datasets. Its cost function, the Kullback–Leibler divergence between two functions of dependence (joint probability density function of components and the product of marginal probability density function of components), allows maintaining the independency among components while increasing dependency of components between datasets (Lee et al., 2008a,b). Based on simulation (Lee et al., 2008b; Dea et al., 2011), IVA shows excellent performance in capturing inter-subject variability and the performance enhancement increases when the spatial variation of a given component across subjects is substantial.

For multimodal imaging analyses, a set of solutions with different emphases have been proposed and extensive reviews of these methods are also available (Biessmann et al., 2011; Sui et al., 2012a). Biessmann et al. (2011) reviewed the multimodal analyses from a variety of perspectives, including multimodal imaging study setup, the advances achieved in basic research and clinical applications, the methods for artifact removal, data-driven and model-driven analyses, and univariate and multivariate fusion. Sui et al. (2012a) focused on comparisons of the multivariate multimodal fusion methods rooted in ICA, canonical component analysis (CCA), and partial least squares (PLS) analysis. Similarity between methods fusing multimodal imaging data and multivariate analyses to bridge imaging and genetics are discussed in the next section.

MULTIVARIATE ANALYSES BRIDGING IMAGING AND GENETICS (CATEGORY 4)

Given the characteristics of imaging and genetic data, multivariate multiple regression is a natural choice, where genetic variants are predictors along with other influencing factors such as age and gender, and imaging variables (regions or voxels of brain) are response variables. In practice with a set of SNPs and brain voxels (they are usually not independent to each other), regularization or modification of traditional multivariate multiple regression has to be taken in place. Wang et al. (2012a) proposed a group sparse regularization on multivariate regression. SNPs are grouped based on genes or LD blocks. A group sparsity to reduce to only genes or LD blocks relevant to all imaging phenotypes, and an individual sparsity to select only important SNPs are all enforced. Lin et al. (2012) presented a projection regression model that is also suitable for imaging genetics. The key of this model is to estimate the principal components of heritability (covariance between multiple phenotypes and genetics of interest), followed by a multivariate regression on the principle components.

When facing a very large number of genetic variants, such as genomic SNPs, and a large number of voxels in the brain, researchers in imaging genetics, very interestingly, has focused on a series of very closely related methods to capture interactive or integrated effects and possibly many genotype-phenotype pairs. These methods include PLS, CCA, reduced rank regression (RRR), and ICA (Hardoon et al., 2009; Liu et al., 2009b; Vounou et al., 2010, 2012; Le Floch et al., 2012; Meda et al., 2012; Chi et al., 2013). They are designed to simultaneously extract latent variables from both genetic and imaging data, which become new genotypes and phenotypes, and the connections of new geno-pheno variables are maximized using different cost functions.

We can use a typical imaging genetic example to illustrate the relation of these methods. We denote by X an $n \times p$ matrix of genetic SNP data, and by Y an $n \times q$ matrix of imaging data, where n is the sample size, p is the size of SNP loci, q is the size of voxels, and $n << p$ or q. The latent variables are obtained through projecting the X or Y to new directions formed by the vectors in U or V matrices. **Figure 2** plots the cost function of each method and the condition under which two different methods become equivalent. PLS maximizes the covariance between latent variables of the two modalities, while CCA maximizes the correlation between them. In a high-dimensional problem where the number of variables is significantly larger than the number of samples, it is common to assume that the covariance matrices of X and Y are diagonal (Vounou et al., 2010; Le Floch et al., 2012). Under such a condition, CCA and PLS become equivalent. The RRR model takes a more general formation that begins from a multivariate linear regression from X to Y, and reduces the rank of the project matrix, a product of UV'. Through minimizing the regression error noted as$(Y - XUV')\Gamma(Y - XUV')'$, RRR obtains the project matrices U and V. When the function of Γ is the identify matrix, RRR is equivalent to PLS, and when the function of Γ is the inverse of covariance matrix $Y'Y$, RRR is equivalent to CCA. Note that the core computations of PLS, CCA and RRR all involve single value decomposition so that the latent variables or projection vectors within one modality (genetic or imaging) are orthogonal to each other. In contrast, ICA emphasizes that latent variables (components) are maximally independent from each other, which can be optimized through many forms of statistical measures, including minimization of mutual information and maximization of non-Gaussianity. One extension of ICA methods applied to imaging genetics is parallel ICA, which simultaneously maximizes both the independence of components and correlations between projection vectors of the two modalities (Liu et al., 2008b).

Parallel ICA was first introduced into imaging genetics in 2009 (Liu et al., 2009b) when applied to a genetic study of schizophrenia

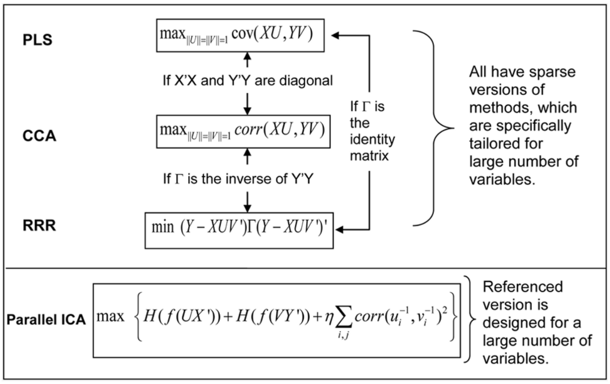

PLS
$\max_{\|U\|=\|V\|=1} \mathrm{cov}(XU, YV)$
If X'X and Y'Y are diagonal
CCA
$\max_{\|U\|=\|V\|=1} corr(XU, YV)$
If Γ is the inverse of Y'Y
RRR
$\min\ (Y - XUV')\Gamma(Y - XUV')'$
If Γ is the identity matrix
All have sparse versions of methods, which are specifically tailored for large number of variables.
Parallel ICA
$\max \left\{ H(f(UX')) + H(f(VY')) + \eta \sum_{i,j} corr(u_i^{-1}, v_i^{-1})^2 \right\}$
Referenced version is designed for a large number of variables.

FIGURE 2 | Cost functions of four multivariate association methods, and their relation and extensions for a large number of variables.

with a 384 SNP array and auditory oddball fMRI data. Since then, this method has been made available for the public through the fusion ICA toolbox[8]. This approach has been utilized by various other groups (Jagannathan et al., 2010; Meda et al., 2010, 2012; Meier et al., 2012). A noteworthy point is that parallel ICA can also be applied onto other types of data in addition to genetics and images (Liu and Calhoun, 2007; Liu et al., 2009a; Wu et al., 2011; Meier et al., 2012). A simulation study showed that parallel ICA performs better within a certain range of sample size vs. genetic variable ratio (Liu et al., 2008a). When a genome-wide high-density large genetic array (e.g., >100K SNP loci) is in place with a relatively small sample size, new extensions of parallel ICA are proposed to improve the performance by incorporating prior information about genetic or imaging data called parallel ICA with reference (Liu et al., 2012a; Chen et al., 2013). As showed by Chen et al. (2013), this approach leverages prior knowledge of known genetic functions to guide ICA for specific components. Thus, a specific SNP factor centered at gene ANK3, which is a schizophrenia susceptibility gene (Ripke et al., 2011), was extracted from a large SNP array (>700K loci). While this method does help extract particular genetic components, which may not be extracted otherwise (Liu et al., 2012a; Chen et al., 2013), its performance relies on the accuracy of reference (Liu et al., 2012a).

As noted above, PLS, CCA, and RRR are closely related. They all introduced the sparse version of algorithms – sparse PLS (Le Floch et al., 2012), sparse CCA (Boutte and Liu, 2010; Chi et al., 2013), and sparse RRR (Vounou et al., 2010, 2012) – when applied onto a large number of variables in imaging genetics. Not only does the increase of sparsity make the interpretation more plausible, but also strengthens the stability of results by avoiding the over-fitting problem. Le Floch et al. (2012) showed through simulation that different levels of regularization on sparsity may produce different results for CCA and PLS, and the two methods converge together with the corresponding regularization strength. Similarly, for RRR, sparsity affects the performance (Vounou et al., 2010), and how to choose sparsity is critical in real applications. Up to now, only sparse PLS (combined with a filtering step) and sparse RRR have been applied to real imaging genetic data with larger than 100k loci (Le Floch et al., 2012; Vounou et al., 2012).

The differences among these methods besides mathematical models listed above also include settings in practice. First, the number of latent variables (components or ranks) to test is chosen differently. CCA, PLS, and RRR extract same numbers of components for genetic and imaging data, and pair-wise connections are tested. Though guidance is discussed for the choice of component number, users of these methods tend to be very conservative. Silver et al. (2012) only investigated the components from first rank in their RRR application, and Vounou et al. (2010, 2012) investigated the top three ranks. In the application of CCA, Hardoon et al. (2009) tested the top pairs of components, and Le Floch et al. (2012) examined the first two pairs of components for both CCA and PLS methods. In contrast, parallel ICA, following the principle of Infomax ICA (Bell and Sejnowski, 1995; Cardoso, 1999), first estimates the number of components embedded in genetic and imaging data. Estimation is either based on information theory (Akaike, 1974; Li et al., 2007) or stability (Chen et al., 2012a), with the goal of reliably, maximally explaining the variance of data. The number of components for genetic and imaging data can be different, and the pairs of related components between the two modalities are driven by data. Sometimes pair-wise correlations are not necessary (Meda et al., 2012). Judging from this aspect, parallel ICA carries advantage of exploring more possible connections between the two modalities, while other methods target only the top correlated components.

Second, all methods are limited in handling a large number of variables (particularly SNP loci). CCA, PLS, and RRR methods may run into over-fitting problems, where cross evaluation performance drops (Le Floch et al., 2012). Parallel ICA fails to identify the connections between modalities (Liu et al., 2008a). The ways to overcome this limitation are also different. Pre-filtering SNP loci to reduce the dimensionality is successfully implemented for CCA and PLS. Le Floch et al. (2012) presented a comprehensive comparison of PLS and CCA combined with different filtering methods. They showed that incorporating a filtering step before the multivariate association test (with the goal of removing irrelevant SNPs) can improve the performance for both methods. Their real data application makes clear that the dimension reduction (which reduced 700k SNPs down to 1000 SNPs) is an important step for avoiding over-fitting with such large genetic data. Although various means can be used to pre-filter SNPs, we recommend leveraging large population genetic data as a reference, such as Psychiatric Genomics Consortium[9]. For RRR, enhancing the sparsity to select only a small number of SNPs is an effective way to increase stability. Yet, the choice of sparsity is not easy (Vounou et al., 2010). N-fold cross evaluation can be used to decide the best parameter. Vounou et al. (2012) chose to test a range of sparsity settings and select resultant SNPs with high probability. Parallel ICA leverages prior information (a referential SNP set) to increase chances of extracting relevant genetic components associated with imaging phenotypes from large SNP data. The difficulty with this approach lies in how to decide the reference. In particular, what we should do when we do not have any prior knowledge about genetics regarding a particular phenotype? While prior information helps interpret the genetic result in a degree, parallel ICA need to threshold the resultant latent variable to select the most weighted SNPs, since no sparsity is in place (Chen et al., 2013).

Third, verification of results from latent variables is very important to guard against false discoveries. N-fold cross evaluation has been utilized for CCA and PLS, and sub-sampling is used in RRR, not exactly verification but increasing the stability (Silver et al., 2012; Vounou et al., 2012). Permutation and leave-one-out evaluation are used in parallel ICA (Liu et al., 2009b; Chen et al., 2012b). We strongly recommend future users to incorporate certain verification steps in their studies, given the complexity of the methods mentioned. To date, only parallel ICA has a ready-to-use package available[10].

[8] http://mialab.mrn.org/software/fit

[9] https://pgc.unc.edu/

[10] http://mialab.mrn.org/software/fit/

Except for multivariate analyses based on latent variables, methods in machine learning category, i.e., training algorithms with known knowledge and using them to predict the unseen data, have also been applied to imaging genetics. For instance, support vector machine on ICA factors of genetic and fMRI data together achieved better separation of schizophrenia patients from controls than using either type of data alone, suggesting that genetic and brain functions capture different, but partially complementary schizophrenic features (Yang et al., 2010). Within the same line, Wang et al. (2012b) proposed a multimodal multitask learning algorithm that combines genetic and multimodal imaging features to predict simultaneously diagnoses and cognitive function. In this algorithm, classification and regression are performed jointly, and a group L1-norm regularization is used for feature selection to integrate heterogeneous imaging genetic data. One of strengths of this approach is that genetic markers and imaging biomarkers relevant for both diagnosis and cognitive function are identified. Another new application of learning algorithms in imaging genetics is random forest on distance matrices, where by employing distance measures between input variables, various interactions (away from original space) are modeled and random forest search is used for selection of best sets of features (Sim et al., 2013). While it provides promising results, the requirement for intensive computation and sophisticated modeling may hinder further applications, which is true for other methods too.

CHALLENGE AND FUTURE DEVELOPMENTS

During the last decade, imaging genetics has rapidly developed into a promising, high impact research field and extended into a body of studies on mental disorders, including both human and animal studies. As Meyer-Lindenberg (2012) stated, future imaging genetic studies have to confront the complexity of epistasis, pleiotropy and gene-by-environment interactions, and this issue will become even more pressing as the field moves into whole genome sequencing. Although methods reviewed here attempt to tackle this complex problem, limitations are clear. For example, none of the methods can really address the genome-wide whole brain association without filtering or dimension reduction. Some multivariate methods such as MDR and prior knowledge guided approaches have not been fully incorporated into imaging genetics yet. Methods of CCA, PLS and RRR, facing over-fitting issues when handling large genetic variables, may be improved by leveraging prior information. Methods of parallel ICA may need to enhance sparsity within the independent genetic components. Such limitation in fact relates to a common problem across multivariate analyses, which is the difficulty in interpreting results (i.e., results are lack of direct biological meaning). For instance, GSEA does not model the exact interaction among SNPs. The latent component does not necessarily hold direct biological reason why multiple genetic variants form into one factor, or why hundreds of voxels group into one brain network. One way to alleviate this problem is to incorporate additional information, such as known biological information, cellular level information, or behavioral specific information, into analyses. Further developing current methods and integrating more information will continue to be an important research frontier.

As matter of fact, another pressing demand raised by Meyer-Lindenberg (2012) in the future of imaging genetics is to integrate various types of data relevant to imaging genetics, beyond just two modalities. The new data can be proteomic, gene expression, epigenetic, behavioral and environmental variables. Studies have shown their relevance to brain structural and functional changes, genetic mutations, and psychiatric disorders (Clark et al., 2006; Serretti et al., 2007; Maric and Svrakic, 2012; Liu et al., 2013). The relationship among these data is by no means simple and pairwise. To date, very few methods have been applied in imagine genetics to tackle the relation beyond two modalities (expect for *post hoc* analyses with behavior or diagnosis). It is very promising to see that some studies have stepped into this direction, though only for multimodal imaging data (Correa et al., 2010; Sui et al., 2012b). How to integrate such data in a systemic way with embedded biological hierarchy is still an untouched land. Methods and models incorporating multiple levels of biological variables (here including behavioral or environmental variables) into broader imaging genetics are another research direction of great potential and impact.

To date, very few studies focused on CNV's effect on brain-based phenotypes (Yeo et al., 2011; Boutte et al., 2012; Liu et al., 2012b), even though many studies have identified a relationship between CNVs with psychiatric disorders (McCarroll and Altshuler, 2007; Bassett et al., 2008; Guilmatre et al., 2009). Meyer-Lindenberg (2012) has indicated that the future of imaging genetics will recognize the importance of the sizeable amount of variation in CNVs. Given the low incidence of individual CNVs, in particular large and rare CNVs, such studies are more likely from multi-site collaborations, where increasing numbers of imaging genetic studies are heading for (Schumann et al., 2010; Thompson et al., 2014). Methods to encompass data from multi-sites, controlling for not only different equipments or experiments but also different local populations or environments, are in great need, which have to consider both computational feasibility and mathematical (model) validity.

Given that the future focus of imaging genetics is expected to be multi-site, large scale, genome-wide whole brain, multiple level association studies, we believe that more effort should be focused on the development of methods that can confront these challenges.

ACKNOWLEDGMENTS

This work was supported by NIH, grant: NIBIB R01EB005846 and P20RR021938 to Vince D. Calhoun, and grant: R33DA027626 to Jingyu Liu.

REFERENCES

Abatangelo, L., Maglietta, R., Distaso, A., D'Addabbo, A., Creanza, T. M., Mukherjee, S., et al. (2009). Comparative study of gene set enrichment methods. *BMC Bioinformatics* 10:275. doi: 10.1186/1471-2105-10-275

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Automat. Contr.* 19, 716–723.

Bach, F., and Michael, J. (2002). Kernel independent component analysis. *J. Mach. Learn. Res.* 3, 1–48.

Bassett, A. S., Marshall, C. R., Lionel, A. C., Chow, E. W., and Scherer, S. W. (2008). Copy number variations and risk for schizophrenia in 22q11.2 deletion syndrome. *Hum. Mol. Genet.* 17, 4045–4053. doi: 10.1093/hmg/ddn307

Beckmann, C. F., and Smith, S. M. (2004). Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans. Med. Imaging* 23, 137–152. doi: 10.1109/TMI.2003.822821

Beckmann, C. F., and Smith, S. M. (2005). Tensorial extensions of independent component analysis for multisubject FMRI analysis. *Neuroimage* 25, 294–311. doi: 10.1016/j.neuroimage.2004.10.043

Bell, A. J., and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.* 7, 1129–1159. doi: 10.1162/neco.1995.7.6.1129

Biessmann, F., Plis, S., Meinecke, F. C., Eichele, T., and Muller, K. R. (2011). Analysis of multimodal neuroimaging data. *IEEE Rev. Biomed. Eng.* 4, 26–58. doi: 10.1109/RBME.2011.2170675

Bigos, K. L., and Weinberger, D. R. (2010). Imaging genetics – days of future past. *Neuroimage* 53, 804–809. doi: 10.1016/j.neuroimage.2010.01.035

Bingham, E., and Hyvarinen, A. (2000). A fast fixed-point algorithm for independent component analysis of complex valued signals. *Int. J. Neural Syst.* 10, 1–8. doi: 10.1142/S0129065700000028

Bookheimer, S. Y., Strojwas, M. H., Cohen, M. S., Saunders, A. M., Pericak-Vance, M. A., Mazziotta, J. C., et al. (2000). Patterns of brain activation in people at risk for Alzheimer's disease. *N. Engl. J. Med.* 343, 450–456. doi: 10.1056/NEJM200008173430701

Boutte, D., Calhoun, V. D., Chen, J., Sabbineni, A., Hutchison, K., and Liu, J. (2012). Association of genetic copy number variations at 11 q14.2 with brain regional volume differences in an alcohol use disorder population. *Alcohol* 46, 519–527. doi: 10.1016/j.alcohol.2012.05.002

Boutte, D., and Liu, J. (2010). "Sparse canonical correlation analysis applied to fMRI and genetic data fusion," in *2010 IEEE International Conference on Bioinformatics and Biomedicine*, Hong Kong, 422–426. doi: 10.1109/BIBM.2010.5706603

Calhoun, V. D., and Adali, T. (2006). Unmixing fMRI with independent component analysis. *IEEE Eng. Med. Biol. Mag.* 25, 79–90. doi: 10.1109/MEMB.2006.1607672

Calhoun, V. D., and Adali, T. (2012). Multisubject independent component analysis of fMRI: a decade of intrinsic networks, default mode, and neurodiagnostic discovery. *IEEE Rev. Biomed. Eng.* 5, 60–73. doi: 10.1109/RBME.2012.2211076

Calhoun, V. D., Adali, T., Pearlson, G. D., and Pekar, J. J. (2001). A method for making group inferences from functional MRI data using independent component analysis. *Hum. Brain Mapp.* 14, 140–151. doi: 10.1002/hbm.1048

Calhoun, V. D., Adali, T., Stevens, M. C., Kiehl, K. A., and Pekar, J. J. (2005). Semi-blind ICA of fMRI: a method for utilizing hypothesis-derived time courses in a spatial ICA analysis. *Neuroimage* 25, 527–538. doi: 10.1016/j.neuroimage.2004.12.012

Calhoun, V. D., Liu, J., and Adali, T. (2009). A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. *Neuroimage* 45, S163–S172. doi: 10.1016/j.neuroimage.2008.10.057

Cardoso, J. F. (1997). Infomax and maximum likelihood for blind source separation. *IEEE Signal Process. Lett.* 4, 112–114. doi: 10.1109/97.566704

Cardoso, J. F. (1999). High-order contrasts for independent component analysis. *Neural Comput.* 11, 157–192. doi: 10.1162/089976699300016863

Cattaert, T., Calle, M. L., Dudek, S. M., Mahachie John, J. M., Van Lishout, F., Urrea, V., et al. (2011). Model-based multifactor dimensionality reduction for detecting epistasis in case-control data in the presence of noise. *Ann. Hum. Genet.* 75, 78–89. doi: 10.1111/j.1469-1809.2010.00604.x

Chen, J., Calhoun, V. D., and Liu, J. (2012a). ICA order selection based on consistency: application to genotype data. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2012, 360–363. doi: 10.1109/EMBC.2012.6345943

Chen, J., Calhoun, V. D., Pearlson, G. D., Ehrlich, S., Turner, J. A., Ho, B. C., et al. (2012b). Multifaceted genomic risk for brain function in schizophrenia. *Neuroimage* 61, 866–875. doi: 10.1016/j.neuroimage.2012.03.022

Chen, J., Calhoun, V. D., Pearlson, G. D., Perrone-Bizzozero, N., Sui, J., Turner, J. A., et al. (2013). Guided exploration of genomic risk for gray matter abnormalities in schizophrenia using parallel independent component analysis with reference. *Neuroimage* 83, 384–396. doi: 10.1016/j.neuroimage.2013.05.073

Chi, E. C., Allen, G. I., Zhou, H., Kohannim, O., Lange, K., and Thompson, P. M. (2013). "Imaging genetics via sparse canonical correlation analysis," in *Biomedical Imaging (ISBI), IEEE 10th International Symposium*, San Francisco, CA. doi: 10.1109/ISBI.2013.6556581

Clark, D., Dedova, I., Cordwell, S., and Matsumoto, I. (2006). A proteome analysis of the anterior cingulate cortex gray matter in schizophrenia. *Mol. Psychiatry* 11, 459–470. doi: 10.1038/sj.mp.4001806

Correa, N. M., Adali, T., Li, Y. O., and Calhoun, V. D. (2010). Canonical correlation analysis for data fusion and group inferences: examining applications of medical imaging data. *IEEE Signal Process. Mag.* 27, 39–50. doi: 10.1109/MSP.2010.936725

Cule, E., Vineis, P., and De Iorio, M. (2011). Significance testing in ridge regression for genetic data. *BMC Bioinformatics* 12:372. doi: 10.1186/1471-2105-12-372

Dea, J. T., Anderson, M., Allen, E., Calhoun, V. D., and Adali, T. (2011). "IVA for multi-subject FMRI analysis: a comparative study using a new simulation toolbox," in *Machine Learning for Signal Processing, IEEE International Workshop*, Beijing, China, 1–6.

Demirci, O., Clark, V. P., and Calhoun, V. D. (2008). A projection pursuit algorithm to classify individuals using fMRI data: application to schizophrenia. *Neuroimage* 39, 1774–1782. doi: 10.1016/j.neuroimage.2007.10.012

D'haeseleer, P. (2005). How does gene expression clustering work? *Nat. Biotechnol.* 23, 1499–1501. doi: 10.1038/nbt1205-1499

Dimitriadou, E., Barth, M., Windischberger, C., Hornik, K., and Moser, E. (2004). A quantitative comparison of functional MRI cluster analysis. *Artif. Intell. Med.* 31, 57–71. doi: 10.1016/j.artmed.2004.01.010

Egan, M. F., Goldberg, T. E., Kolachana, B. S., Callicott, J. H., Mazzanti, C. M., Straub, R. E., et al. (2001). Effect of COMT Val108/158 Met genotype on frontal lobe function and risk for schizophrenia. *Proc. Natl. Acad. Sci. U.S.A.* 98, 6917–6922. doi: 10.1073/pnas.111134598

Engreitz, J. M., Daigle, B. J. Jr., Marshall, J. J., and Altman, R. B. (2010). Independent component analysis: mining microarray data for fundamental human gene expression modules. *J. Biomed. Inform.* 43, 932–944. doi: 10.1016/j.jbi.2010.07.001

Erhardt, E. B., Rachakonda, S., Bedrick, E. J., Allen, E. A., Adali, T., and Calhoun, V. D. (2011). Comparison of multi-subject ICA methods for analysis of fMRI data. *Hum. Brain Mapp.* 32, 2075–2095. doi: 10.1002/hbm.21170

Esposito, F., Scarabino, T., Hyvarinen, A., Himberg, J., Formisano, E., Comani, S., et al. (2005). Independent component analysis of fMRI group studies by self-organizing clustering. *Neuroimage* 25, 193–205. doi: 10.1016/j.neuroimage.2004.10.042

Filipovych, R., and Davatzikos, C. (2011). Semi-supervised pattern classification of medical images: application to mild cognitive impairment (MCI). *Neuroimage* 55, 1109–1119. doi: 10.1016/j.neuroimage.2010.12.066

Glahn, D. C., Thompson, P. M., and Blangero, J. (2007). Neuroimaging endophenotypes: strategies for finding genes influencing brain structure and function. *Hum. Brain Mapp.* 28, 488–501. doi: 10.1002/hbm.20401

Gottesman, I. I., and Gould, T. D. (2003). The endophenotype concept in psychiatry: etymology and strategic intentions. *Am. J. Psychiatry* 160, 636–645. doi: 10.1176/appi.ajp.160.4.636

Gui, J., Andrew, A. S., Andrews, P., Nelson, H. M., Kelsey, K. T., Karagas, M. R., et al. (2011). A robust multifactor dimensionality reduction method for detecting gene-gene interactions with application to the genetic analysis of bladder cancer susceptibility. *Ann. Hum. Genet.* 75, 20–28. doi: 10.1111/j.1469-1809.2010.00624.x

Gui, J., Moore, J. H., Williams, S. M., Andrews, P., Hillege, H. L., Van Der Harst, P., et al. (2013). A simple and computationally efficient approach to multifactor dimensionality reduction analysis of gene-gene interactions for quantitative traits. *PLoS ONE* 8:e66545. doi: 10.1371/journal.pone.0066545

Guilmatre, A., Dubourg, C., Mosca, A. L., Legallic, S., Goldenberg, A., Drouin-Garraud, V., et al. (2009). Recurrent rearrangements in synaptic and neurodevelopmental genes and shared biologic pathways in schizophrenia, autism, and mental retardation. *Arch. Gen. Psychiatry* 66, 947–956. doi: 10.1001/archgenpsychiatry.2009.80

Hardoon, D. R., Ettinger, U., Mourao-Miranda, J., Antonova, E., Collier, D., Kumari, V., et al. (2009). Correlation-based multivariate analysis of genetic influence on brain volume. *Neurosci. Lett.* 450, 281–286. doi: 10.1016/j.neulet.2008.11.035

Hariri, A. R., Drabant, E. M., and Weinberger, D. R. (2006). Imaging genetics: perspectives from studies of genetically driven variation in serotonin function and corticolimbic affective processing. *Biol. Psychiatry* 59, 888–897. doi: 10.1016/j.biopsych.2005.11.005

Heinz, A., Goldman, D., Jones, D. W., Palmour, R., Hommer, D., Gorey, J. G., et al. (2000). Genotype influences *in vivo* dopamine transporter availability in human striatum. *Neuropsychopharmacology* 22, 133–139. doi: 10.1016/S0893-133X(99)00099-8

Hinrichs, C., Singh, V., Mukherjee, L., Xu, G., Chung, M. K., and Johnson, S. C. (2009). Spatially augmented LPboosting for AD classification with evaluations on the ADNI dataset. *Neuroimage* 48, 138–149. doi: 10.1016/j.neuroimage.2009.05.056

Holden, M., Deng, S., Wojnowski, L., and Kulle, B. (2008). GSEA-SNP: applying gene set enrichment analysis to SNP data from genome-wide association studies. *Bioinformatics* 24, 2784–2785. doi: 10.1093/bioinformatics/btn516

Hong, B., Pearlson, G. D., and Calhoun, V. D. (2005). Source density-driven independent component analysis approach for fMRI data. *Hum. Brain Mapp.* 25, 297–307. doi: 10.1002/hbm.20100

Hu, T., Sinnott-Armstrong, N. A., Kiralis, J. W., Andrew, A. S., Karagas, M. R., and Moore, J. H. (2011). Characterizing genetic interactions in human disease association studies using statistical epistasis networks. *BMC Bioinformatics* 12:364. doi: 10.1186/1471-2105-12-364

Huang, D. S., and Zheng, C. H. (2006). Independent component analysis-based penalized discriminant method for tumor classification using gene expression data. *Bioinformatics* 22, 1855–1862. doi: 10.1093/bioinformatics/btl190

Hyvirinen, A., and Oja, E. (1999). A fast fixed-point algorithm for independent component analysis. *Neural Comput.* 9, 1483–1492. doi: 10.1162/neco.1997.9.7.1483

Jagannathan, K., Calhoun, V. D., Gelernter, J., Stevens, M. C., Liu, J., Bolognani, F., et al. (2010). Genetic associations of brain structural networks in schizophrenia: a preliminary study. *Biol. Psychiatry* 68, 657–666. doi: 10.1016/j.biopsych.2010.06.002

Jiang, D., Tang, C., and Zahng, A. (2004). Cluster analysis for gene expression data: a survey. *IEEE Trans. Knowl. Data Eng.* 16, 1370–1386. doi: 10.1109/TKDE.2004.68

Jombart, T., Pontier, D., and Dufour, A. B. (2009). Genetic markers in the playground of multivariate analysis. *Heredity* 102, 330–341. doi: 10.1038/hdy.2008.130

Kim, T., Lee, I., and Lee, T.-W. (2006). "Independent vector analysis: definition and algorithms," in *Signals, Systems and Computers, ACSSC '06. Fortieth Asilomar Conference on*, Pacific Grove, CA, 1393–1396. doi: 10.1109/ACSSC.2006.354986

Kong, W., Vanderburg, C. R., Gunshin, H., Rogers, J. T., and Huang, X. (2008). A review of independent component analysis application to microarray gene expression data. *Biotechniques* 45, 501–520. doi: 10.2144/000112950

Lee, J. H., Lee, T. W., Jolesz, F. A., and Yoo, S. S. (2008a). Independent vector analysis (IVA) for group fMRI processing of subcortical area. *Int. J. Imaging Syst. Tech.* 18, 29–41. doi: 10.1002/ima.20141

Lee, J. H., Lee, T. W., Jolesz, F. A., and Yoo, S. S. (2008b). Independent vector analysis (IVA): multivariate approach for fMRI group study. *Neuroimage* 40, 86–109. doi: 10.1016/j.neuroimage.2007.11.019

Le Floch, E., Guillemot, V., Frouin, V., Pinel, P., Lalanne, C., Trinchera, L., et al. (2012). Significant correlation between a set of genetic polymorphisms and a functional brain network revealed by feature selection and sparse Partial Least Squares. *Neuroimage* 63, 11–24. doi: 10.1016/j.neuroimage.2012.06.061

Li, M. X., Gui, H. S., Kwan, J. S., and Sham, P. C. (2011). GATES: a rapid and powerful gene-based association test using extended simes procedure. *Am. J. Hum. Genet.* 88, 283–293. doi: 10.1016/j.ajhg.2011.01.019

Li, Y. O., Adali, T., and Calhoun, V. D. (2007). Estimating the number of independent components for functional magnetic resonance imaging data. *Hum. Brain Mapp.* 28, 1251–1266. doi: 10.1002/hbm.20359

Liu, J., Bixler, J. N., and Calhoun, V. D. (2008a). "A multimodality ICA study – integrating genomic single nucleotide polymorphisms with functional neuroimaging data," in *Bioinformatics and Biomedicine Workshops, 2008. BIBMW 2008* (IEEE International Conference on), Philadelphia, PA, 151–157. doi: 10.1109/BIBMW.2008.4686229

Liu, J., and Calhoun, V. (2007). "Parallel independent component analysis for multimodel analysis: application to fMRI and EEG data," *in Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, Washington, DC, 1028–1031. doi: 10.1109/ISBI.2007.357030

Liu, J., Chen, J., Ehrlich, S., Walton, E., White, T., Perrone-Bizzozero, N., et al. (2013). Methylation patterns in whole blood correlate with symptoms in schizophrenia patients. *Schizophr. Bull.* doi: 10.1093/schbul/sbt080 [Epub ahead of print].

Liu, J., Demirci, O., and Calhoun, V. D. (2008b). A parallel independent component analysis approach to investigate genomic influence on brain function. *IEEE Signal Process. Lett.* 15, 413–416. doi: 10.1109/LSP.2008.922513

Liu, J., Ghassemi, M. M., Michael, A. M., Boutte, D., Wells, W., Perrone-Bizzozero, N., et al. (2012a). An ICA with reference approach in identification of genetic variation and associated brain networks. *Front. Hum. Neurosci.* 6:21. doi: 10.3389/fnhum.2012.00021

Liu, J., Hutchison, K., Perrone-Bizzozero, N., Morgan, M., Sui, J., and Calhoun, V. (2010a). Identification of genetic and epigenetic marks involved in population structure. *PLoS ONE* 5:e13209. doi: 10.1371/journal.pone.0013209

Liu, J., Kiehl, K. A., Pearlson, G., Perrone-Bizzozero, N. I., Eichele, T., and Calhoun, V. D. (2009a). Genetic determinants of target and novelty-related event-related potentials in the auditory oddball response. *Neuroimage* 46, 809–816. doi: 10.1016/j.neuroimage.2009.02.045

Liu, J., Pearlson, G., Windemuth, A., Ruano, G., Perrone-Bizzozero, N. I., and Calhoun, V. (2009b). Combining fMRI and SNP data to investigate connections between brain function and genetics using parallel ICA. *Hum. Brain Mapp.* 30, 241–255. doi: 10.1002/hbm.20508

Liu, J., Ulloa, A., Perrone-Bizzozero, N., Yeo, R., Chen, J., and Calhoun, V. D. (2012b). A pilot study on collective effects of 22q13.31 deletions on gray matter concentration in schizophrenia. *PLoS ONE* 7:e52865. doi: 10.1371/journal.pone.0052865

Lin, J. A., Zhu, H., Knickmeyer, R., Styner, M., Gilmore, J., and Ibrahim, J. G. (2012). Projection regression models for multivariate imaging phenotype. *Genet. Epidemiol.* 36, 631–641. doi: 10.1002/gepi.21658

Liu, J. Z., Mcrae, A. F., Nyholt, D. R., Medland, S. E., Wray, N. R., Brown, K. M., et al. (2010b). A versatile gene-based test for genome-wide association studies. *Am. J. Hum. Genet.* 87, 139–145. doi: 10.1016/j.ajhg.2010.06.009

Lin, Q. H., Liu, J., Zheng, Y. R., Liang, H., and Calhoun, V. D. (2010). Semiblind spatial ICA of fMRI using spatial constraints. *Hum. Brain Mapp.* 31, 1076–1088. doi: 10.1002/hbm.20919

Lou, X. Y., Chen, G. B., Yan, L., Ma, J. Z., Zhu, J., Elston, R. C., et al. (2007). A generalized combinatorial approach for detecting gene-by-gene and gene-by-environment interactions with application to nicotine dependence. *Am. J. Hum. Genet.* 80, 1125–1137. doi: 10.1086/518312

Ma, D. Q., Whitehead, P. L., Menold, M. M., Martin, E. R., Ashley-Koch, A. E., Mei, H., et al. (2005). Identification of significant association and gene-gene interaction of GABA receptor subunit genes in autism. *Am. J. Hum. Genet.* 77, 377–388. doi: 10.1086/433195

Maric, N. P., and Svrakic, D. M. (2012). Why schizophrenia genetics needs epigenetics: a review. *Psychiatr. Danub.* 24, 2–18.

McCarroll, S. A., and Altshuler, D. M. (2007). Copy-number variation and association studies of human disease. *Nat. Genet.* 39, S37–S42. doi: 10.1038/ng2080

McKeown, M. J., Hansen, L. K., and Sejnowsk, T. J. (2003). Independent component analysis of functional MRI: what is signal and what is noise? *Curr. Opin. Neurobiol.* 13, 620–629. doi: 10.1016/j.conb.2003.09.012

Meda, S. A., Jagannathan, K., Gelernter, J., Calhoun, V. D., Liu, J., Stevens, M. C., et al. (2010). A pilot multivariate parallel ICA study to investigate differential linkage between neural networks and genetic profiles in schizophrenia. *Neuroimage* 53, 1007–1015. doi: 10.1016/j.neuroimage.2009.11.052

Meda, S. A., Narayanan, B., Liu, J., Perrone-Bizzozero, N. I., Stevens, M. C., Calhoun, V. D., et al. (2012). A large scale multivariate parallel ICA method reveals novel imaging-genetic relationships for Alzheimer's disease in the ADNI cohort. *Neuroimage* 60, 1608–1621. doi: 10.1016/j.neuroimage.2011.12.076

Meier, T. B., Wildenberg, J. C., Liu, J., Chen, J., Calhoun, V. D., Biswal, B. B., et al. (2012). Parallel ICA identifies sub-components of resting state networks that covary with behavioral indices. *Front. Hum. Neurosci.* 6:281. doi: 10.3389/fnhum.2012.00281

Meyer-Lindenberg, A. (2010). Imaging genetics of schizophrenia. *Dialogues Clin. Neurosci.* 12, 449–456.

Meyer-Lindenberg, A. (2012). The future of fMRI and genetics research. *Neuroimage* 62, 1286–1292. doi: 10.1016/j.neuroimage.2011.10.063

Meyer-Lindenberg, A., Nicodemus, K. K., Egan, M. F., Callicott, J. H., Mattay, V., and Weinberger, D. R. (2008). False positives in imaging genetics. *Neuroimage* 40, 655–661. doi: 10.1016/j.neuroimage.2007.11.058

Meyer-Lindenberg, A., and Weinberger, D. R. (2006). Intermediate phenotypes and genetic mechanisms of psychiatric disorders. *Nat. Rev. Neurosci.* 7, 818–827. doi: 10.1038/nrn1993

Mirina, A., Atzmon, G., Ye, K., and Bergman, A. (2012). Gene size matters. *PLoS ONE* 7:e49093. doi: 10.1371/journal.pone.0049093

Moore, J. H., Asselbergs, F. W., and Williams, S. M. (2010). Bioinformatics challenges for genome-wide association studies. *Bioinformatics* 26, 445–455. doi: 10.1093/bioinformatics/btp713

Moore, J. H., and Williams, S. M. (2002). New strategies for identifying gene-gene interactions in hypertension. *Ann. Med.* 34, 88–95. doi: 10.1080/07853890252953473

Mootha, V. K., Lindgren, C. M., Eriksson, K. F., Subramanian, A., Sihag, S., Lehar, J., et al. (2003). PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.* 34, 267–273. doi: 10.1038/ng1180

Oh, S., Lee, J., Kwon, M. S., Weir, B., Ha, K., and Park, T. (2012). A novel method to identify high order gene-gene interactions in genome-wide association studies: gene-based MDR. *BMC Bioinformatics* 13(Suppl. 9):S5. doi: 10.1186/1471-2105-13-S9-S5

Oh, S. J., Ahn, J. Y., and Chung, D. H. (2011). Comparison of invariant NKT cells with conventional T cells by using gene set enrichment analysis (GSEA). *Immune Netw.* 11, 406–411. doi: 10.4110/in.2011.11.6.406

Pan, Q., Hu, T., and Moore, J. H. (2013). Epistasis, complexity, and multifactor dimensionality reduction. *Methods Mol. Biol.* 1019, 465–477. doi: 10.1007/978-1-62703-447-0-22

Papassotiropoulos, A., and de Quervain, D. J. (2011). Genetics of human episodic memory: dealing with complexity. *Trends Cogn. Sci.* 15, 381–387. doi: 10.1016/j.tics.2011.07.005

Pattin, K. A., White, B. C., Barney, N., Gui, J., Nelson, H. H., Kelsey, K. T., et al. (2009). A computationally efficient hypothesis testing method for epistasis analysis using multifactor dimensionality reduction. *Genet. Epidemiol.* 33, 87–94. doi: 10.1002/gepi.20360

Potkin, S. G., Guffanti, G., Lakatos, A., Turner, J. A., Kruggel, F., Fallon, J. H., et al. (2009). Hippocampal atrophy as a quantitative trait in a genome-wide association study identifying novel susceptibility genes for Alzheimer's disease. *PLoS ONE* 4:e6501. doi: 10.1371/journal.pone.0006501

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.* 38, 904–909. doi: 10.1038/ng1847

Purcell, S. M., Wray, N. R., Stone, J. L., Visscher, P. M., O'donovan, M. C., Sullivan, P. F., et al. (2009). Common polygenic variation contributes to risk of schizophrenia and bipolar disorder. *Nature* 460, 748–752. doi: 10.1038/nature08185

Rasetti, R., and Weinberger, D. R. (2011). Intermediate phenotypes in psychiatric disorders. *Curr. Opin. Genet. Dev.* 21, 340–348. doi: 10.1016/j.gde.2011.02.003

Ripke, S., Sanders, A. R., Kendler, K. S., Levinson, D. F., Sklar, P., Holmans, P. A., et al. (2011). Genome-wide association study identifies five new schizophrenia loci. *Nat. Genet.* 43, 969–976. doi: 10.1038/ng.940

Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., et al. (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *Am. J. Hum. Genet.* 69, 138–147. doi: 10.1086/321276

Roffman, J. L., Weiss, A. P., Goff, D. C., Rauch, S. L., and Weinberger, D. R. (2006). Neuroimaging-genetic paradigms: a new approach to investigate the pathophysiology and treatment of cognitive deficits in schizophrenia. *Harv. Rev. Psychiatry* 14, 78–91. doi: 10.1080/10673220600642945

Rose, E. J., and Donohoe, G. (2013). Brain vs behavior: an effect size comparison of neuroimaging and cognitive studies of genetic risk for schizophrenia. *Schizophr. Bull.* 39, 518–526. doi: 10.1093/schbul/sbs056

Saidi, S. A., Holland, C. M., Kreil, D. P., Mackay, D. J., Charnock-Jones, D. S., Print, C. G., et al. (2004). Independent component analysis of microarray data in the study of endometrial cancer. *Oncogene* 23, 6677–6683. doi: 10.1038/sj.onc.1207562

Schmithorst, V. J., and Holland, S. K. (2004). Comparison of three methods for generating group statistical inferences from independent component analysis of functional magnetic resonance imaging data. *J. Magn. Reson. Imaging* 19, 365–368. doi: 10.1002/jmri.20009

Schumann, G., Loth, E., Banaschewski, T., Barbot, A., Barker, G., Buchel, C., et al. (2010). The IMAGEN study: reinforcement-related behaviour in normal brain function and psychopathology. *Mol. Psychiatry* 15, 1128–1139. doi: 10.1038/mp.2010.4

Serretti, A., Olgiati, P., and De Ronchi, D. (2007). Genetics of Alzheimer's disease. a rapidly evolving field. *J. Alzheimers Dis.* 12, 73–92.

Sheng, J., Deng, H. W., Calhoun, V. D., and Wang, Y. P. (2011). Integrated analysis of gene expression and copy number data on gene shaving using independent component analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 1568–1579. doi: 10.1109/TCBB.2011.71

Silver, M., Janousova, E., Hua, X., Thompson, P. M., and Montana, G. (2012). Identification of gene pathways implicated in Alzheimer's disease using longitudinal imaging phenotypes with sparse regression. *Neuroimage* 63, 1681–1694. doi: 10.1016/j.neuroimage.2012.08.002

Silver, M., Montana, G., and Nichols, T. E. (2011). False positives in neuroimaging genetics using voxel-based morphometry data. *Neuroimage* 54, 992–1000. doi: 10.1016/j.neuroimage.2010.08.049

Sim, A., Tsagkrasoulis, D., and Montana, G. (2013). Random forests on distance matrices for imaging genetics studies. *Stat. Appl. Genet. Mol. Biol.* 12, 757–786. doi: 10.1515/sagmb-2013-0040

Sivakumaran, S., Agakov, F., Theodoratou, E., Prendergast, J. G., Zgaga, L., Manolio, T., et al. (2011). Abundant pleiotropy in human complex diseases and traits. *Am. J. Hum. Genet.* 89, 607–618. doi: 10.1016/j.ajhg.2011.10.004

Sloan, C. D., Shen, L., West, J. D., Wishart, H. A., Flashman, L. A., Rabin, L. A., et al. (2010). Genetic pathway-based hierarchical clustering analysis of older adults with cognitive complaints and amnestic mild cognitive impairment using clinical and neuroimaging phenotypes. *Am. J. Med. Genet. B Neuropsychiatr. Genet.* 153B, 1060–1069. doi: 10.1002/ajmg.b.31078

Small, G. W., Ercoli, L. M., Silverman, D. H., Huang, S. C., Komo, S., Bookheimer, S. Y., et al. (2000). Cerebral metabolic and cognitive decline in persons at genetic risk for Alzheimer's disease. *Proc. Natl. Acad. Sci. U.S.A.* 97, 6037–6042. doi: 10.1073/pnas.090106797

Smoller, J. W., Craddock, N., Kendler, K., Lee, P. H., Neale, B. M., Nurnberger, J. I., et al. (2013). Identification of risk loci with shared effects on five major psychiatric disorders: a genome-wide analysis. *Lancet* 381, 1371–1379. doi: 10.1016/S0140-6736(12)62129-1

Stein, J. L., Hua, X., Lee, S., Ho, A. J., Leow, A. D., Toga, A. W., et al. (2010). Voxelwise genome-wide association study (vGWAS). *Neuroimage* 53, 1160–1174. doi: 10.1016/j.neuroimage.2010.02.032

Suarez-Farinas, M., Lowes, M. A., Zaba, L. C., and Krueger, J. G. (2010). Evaluation of the psoriasis transcriptome across different studies by gene set enrichment analysis (GSEA). *PLoS ONE* 5:e10247. doi: 10.1371/journal.pone.0010247

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.* 102, 15545–15550. doi: 10.1073/pnas.0506580102

Sui, J., Adali, T., Yu, Q., Chen, J., and Calhoun, V. D. (2012a). A review of multivariate methods for multimodal fusion of brain imaging data. *J. Neurosci. Methods* 204, 68–81. doi: 10.1016/j.jneumeth.2011.10.031

Sui, J., He, H., Pearlson, G. D., Adali, T., Kiehl, K. A., Yu, Q., et al. (2012b). Three-way (N-way) fusion of brain imaging data based on mCCA + jICA and its application to discriminating schizophrenia. *Neuroimage* 66C, 119–132. doi: 10.1016/j.neuroimage.2012.10.051

Thompson, P. M., Stein, J. L., Medland, S. E., Hibar, D. P., Vasquez, A. A., Renteria, M. E., et al. (2014). The ENIGMA consortium: large-scale collaborative analyses of neuroimaging and genetic data. *Brain Imaging Behav.* doi: 10.1007/s11682-013-9269-5 [Epub ahead of print].

Vounou, M., Janousova, E., Wolz, R., Stein, J. L., Thompson, P. M., Rueckert, D., et al. (2012). Sparse reduced-rank regression detects genetic associations with voxel-wise longitudinal phenotypes in Alzheimer's disease. *Neuroimage* 60, 700–716. doi: 10.1016/j.neuroimage.2011.12.029

Vounou, M., Nichols, T. E., and Montana, G. (2010). Discovering genetic associations with high-dimensional neuroimaging phenotypes: a sparse reduced-rank regression approach. *Neuroimage* 53, 1147–1159. doi: 10.1016/j.neuroimage.2010.07.002

Walton, E., Turner, J., Gollub, R. L., Manoach, D. S., Yendiki, A., Ho, B. C., et al. (2013). Cumulative genetic risk and prefrontal activity in patients with schizophrenia. *Schizophr. Bull.* 39, 703–711. doi: 10.1093/schbul/sbr190

Wang, K., and Abbott, D. (2008). A principal components regression approach to multilocus genetic association studies. *Genet. Epidemiol.* 32, 108–118. doi: 10.1002/gepi.20266

Wang, H., Nie, F., Huang, H., Kim, S., Nho, K., Risacher, S. L., et al. (2012a). Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the ADNI cohort. *Bioinformatics* 28, 229–237. doi: 10.1093/bioinformatics/btr649

Wang, H., Nie, F., Huang, H., Risacher, S. L., Saykin, A. J., and Shen, L. (2012b). Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning. *Bioinformatics* 28, i127–i136. doi: 10.1093/bioinformatics/bts228

Weng, L., Macciardi, F., Subramanian, A., Guffanti, G., Potkin, S. G., Yu, Z., et al. (2011). SNP-based pathway enrichment analysis for genome-wide association studies. *BMC Bioinformatics* 12:99. doi: 10.1186/1471-2105-12-99

Whalley, H. C., Papmeyer, M., Sprooten, E., Romaniuk, L., Blackwood, D. H., Glahn, D. C., et al. (2012). The influence of polygenic risk for bipolar disorder on neural activation assessed using fMRI. *Trans. Psychiatry* 2, e130. doi: 10.1038/tp.2012.60

Winham, S. (2013). Applications of multifactor dimensionality reduction to genome-wide data using the R package "MDR." *Methods Mol. Biol.* 1019, 479–498. doi: 10.1007/978-1-62703-447-0-23

Wu, L., Eichele, T., and Calhoun, V. (2011). "Parallel independent component analysis using an optimized neurovascular coupling for concurrent EEG-fMRI sources," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, Massachusetts, 2542–2545.

Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics* 25, 714–721. doi: 10.1093/bioinformatics/btp041

Yang, H., Liu, J., Sui, J., Pearlson, G., and Calhoun, V. D. (2010). A hybrid machine learning method for fusing fMRI and genetic data: combining both improves classification of schizophrenia. *Front. Hum. Neurosci.* 4:192. doi: 10.3389/fnhum.2010.00192

Yeo, R. A., Gangestad, S. W., Gasparovic, C., Liu, J., Calhoun, V. D., Thoma, R. J., et al. (2011). Rare copy number deletions predict individual variation in human brain metabolite concentrations in individuals with alcohol use disorders. *Biol. Psychiatry*. 70, 537–544. doi: 10.1016/j.biopsych.2011.04.019

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 31 October 2013; accepted: 04 March 2014; published online: 26 March 2014.
Citation: Liu J and Calhoun VD (2014) A review of multivariate analyses in imaging genetics. Front. Neuroinform. ***8****:29. doi: 10.3389/fninf.2014.00029*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

REVIEW ARTICLE
published: 26 February 2014
doi: 10.3389/fninf.2014.00019

Neuroinformatics challenges to the structural, connectomic, functional, and electrophysiological multimodal imaging of human traumatic brain injury

S. Y. Matthew Goh, Andrei Irimia, Carinna M. Torgerson and John D. Van Horn*

Department of Neurology, Institute for Neuroimaging and Informatics, Keck School of Medicine, University of Southern California, Los Angeles, CA, USA

Edited by:
Qian Luo, Saint Louis University, USA

Reviewed by:
Gully A. P. C. Burns, USC Information Sciences Institute, USA
Jeffrey Stout, Illinois Institute of Technology, USA

****Correspondence:***
John D. Van Horn, Department of Neurology, Institute for Neuroimaging and Informatics, Keck School of Medicine, University of Southern California, 2001 North Soto Street, Room 102, Los Angeles, CA 90089-9235, USA
e-mail: jvanhorn@usc.edu

Throughout the past few decades, the ability to treat and rehabilitate traumatic brain injury (TBI) patients has become critically reliant upon the use of neuroimaging to acquire adequate knowledge of injury-related effects upon brain function and recovery. As a result, the need for TBI neuroimaging analysis methods has increased in recent years due to the recognition that spatiotemporal computational analyses of TBI evolution are useful for capturing the effects of TBI dynamics. At the same time, however, the advent of such methods has brought about the need to analyze, manage, and integrate TBI neuroimaging data using informatically inspired approaches which can take full advantage of their large dimensionality and informational complexity. Given this perspective, we here discuss the neuroinformatics challenges for TBI neuroimaging analysis in the context of structural, connectivity, and functional paradigms. Within each of these, the availability of a wide range of neuroimaging modalities can be leveraged to fully understand the heterogeneity of TBI pathology; consequently, large-scale computer hardware resources and next-generation processing software are often required for efficient data storage, management, and analysis of TBI neuroimaging data. However, each of these paradigms poses challenges in the context of informatics such that the ability to address them is critical for augmenting current capabilities to perform neuroimaging analysis of TBI and to improve therapeutic efficacy.

Keywords: neuroinformatics, traumatic brain injury, neuroanatomy, connectomics, rehabilitation, MRI, DTI

INTRODUCTION

Traumatic brain injury (TBI) affects ~1.7 million people in the United States every year, leading to roughly 50,000 cases of mortality and 80,000 cases of permanent severe neurological disability annually (Ghajar, 2000; Faul et al., 2010). Throughout the past few decades, the use of neuroimaging to acquire knowledge of injury-related effects upon brain function and recovery has become prominent due to the recognition that spatiotemporal computational analyses of TBI evolution are useful for capturing the effects of its dynamics (Irimia et al., 2011). On the other hand, the proliferation of neuroimaging studies has brought about the need to analyze, manage, and integrate TBI neuroimaging data with sophisticated neuroinformatics methods which can address and handle their large dimensionality and informational complexity.

The high dimensionality of TBI neuroimaging data poses one of the most significant challenges to the development and implementation of data processing workflows for TBI analysis. This dimensionality stems partly from the fact that various types of magnetic resonance imaging (MRI) sequences reveal only certain aspects of TBI pathology, which implies that their combined use is often necessary in order to acquire a comprehensive view of TBI lesion type and extent. For instance, fluid attenuated inversion recovery (FLAIR) and susceptibility weighted imaging (SWI) are MRI sequence types which are suitable for detecting edema and cerebral micro-hemorrhages, respectively (Irimia et al., 2011). Partly because of such qualitative and quantitative differences between MRI sequence types as well as between MRI and other imaging modalities such as computed tomography (CT) and positron emission tomography (PET), a vital component of TBI neuroimaging involves the availability of multimodal neuroimaging data sets to aid in the identification and characterization of pathology.

Present methodologies for long-term clinical assessment of this condition include the use of scoring scales such as the Glasgow Coma Scale (GCS), which is a frequently used evaluator of consciousness level and head injury severity. Additional clinical measures of functional outcome after TBI which are used in clinical practice include acute physiology and chronic health evaluation (APACHE), Mortality Probability Model (MPM), and simplified acute physiology score (SAPS; Vincent and Moreno, 2010), all of which can be complemented by neuroimaging-based metrics. In the case of the GCS and of other currently available scoring systems, their effectiveness in providing prognostic information is hampered by their limited descriptiveness. By contrast, computational analyses of multimodal structural neuroimaging data offer a variety of ways in which pathological changes can be assessed. It is important to note that (a) the GCS is typically used in conjunction with a number of other clinical measures and physiological metrics, and that (b) computational analyses vs. clinical scoring systems fulfill different roles. Thus, one theme of this review is that the drawbacks of conventional clinical scoring systems can be complemented by outcome prediction models formulated using neuroinformatics tools, which require the exploration and mining of quantitative metrics derived from structural neuroimaging data.

Given the current trends in TBI neuroimaging, this review aims to highlight and draw the attention of the neuroinformatics community to the challenges encountered in the study of human TBI within the context of three distinct types of neuroimaging: structural, connectivity, and functional. It attempts to suggest how novel data-driven solutions should be formulated to assist TBI neuroimaging analysis with the ultimate purpose of improving therapeutic efficacy. The analytic approaches examined below outline the use of a varied number of neuroimaging techniques and demonstrate the wealth of knowledge obtainable through quantitative analysis of neuroimaging data. We propose that, to improve rehabilitation strategies and the accuracy of TBI patient outcome prediction, it is necessary to augment existing capabilities to facilitate the multimodal use of neuroimaging methods and of their application to large population samples of TBI patients, as well as to individual patients by means of personalized approaches. This task should be reliant on continued development, support and input from neurologists, neuroinformaticians and biostatisticians to provide the theoretical tools and practical mechanisms required for technological and scientific progress in this field of high priority to public health.

STRUCTURAL NEUROIMAGING APPROACHES

Computational methods for the analysis of brain structure provide a powerful approach to the investigation of TBI-related pathology. Typical quantitative metrics for the study of brain structure include morphometric measures (e.g., the curvature and folding index of the cortex) and volumetric measures – e.g., cortical thickness, gray matter (GM) volume, white matter (WM) volume, etc. – which have been highly useful in describing neuroanatomical profiles at the macroscopic level, in both health and in a variety of pathological conditions (Ashburner and Friston, 2000; Thompson et al., 2003). One motivating factor behind the decision to undertake the calculations of these metrics within large collaborative efforts such as the Alzheimer's Disease (AD) Neuroimaging Initiative (ADNI) has been the desire to identify biomarkers which are prognostic and informative of clinical outcome, and which can be used to optimize the formulation of patient treatment as well as the selection of rehabilitation protocols, as in the study of Jack et al. (2008). The latter authors aimed to address the neuroinformatics challenge of longitudinal ADNI data processing by (a) linking all data at each time point, (b) making a repository available to the scientific community, (c) developing technical standards for longitudinal imaging studies, (d) determining optimum methods for image acquisition and analysis, and (e) validating imaging biomarker data. Such goals are excellently suited for future human TBI studies as well. All of these tasks involve neuroinformatics approaches which are currently insufficiently available in human TBI research. Subsequently, the ability to perform relevant systematic and quantitative analyses of TBI brain structure has been appreciably affected by a number of formidable challenges which this section aims to highlight.

One challenge encountered during the task of constructing TBI data analysis workflows for the extraction of clinically relevant information is the task of tissue segmentation, a process often associated with the three-dimensional analysis of MRI volumes. In neuroimaging, tissue segmentation refers to the classification of voxels from MRI data into relevant tissue types (e.g., GM, WM, cerebrospinal fluid, non-cortical structures) so that morphometric and volumetric measures can be quantified. Typically, tissue segmentation is a complex procedure involving the correction of magnetic field inhomogeneities, image intensity normalization, extra-cerebral voxel removal via skull-stripping, and the assignment of each voxel to one of several classes (WM, GM, etc.) using a probabilistic model based on image intensity differences between voxels belonging to each class (Dale et al., 1999). Though there are a wide variety of approaches to segmentation including those based on machine learning (Powell et al., 2008; Hofmann et al., 2011), brain tissue segmentation often incorporates the application of anatomical priors while computing the probability of a voxel belonging to a certain tissue type (Irimia et al., 2011). Whereas the application of such anatomical priors is typically quite feasible in the case of healthy brains, this class of methods is known to fail when applied to moderate or severe TBI volumes because, in such cases, (a) TBI neuroanatomy can differ substantially from health due to the presence of gross pathology and (b) edema and hemorrhage can dramatically alter voxel intensities, thereby modifying the spatial mapping of such voxels to atlas space in an undesirable manner. Thus, segmentation of TBI volumes can be particularly difficult to automate due to the heterogeneity of injury location, shape, and size, none of which are easily predictable (Filippi et al., 1998). Nevertheless, it is important to acknowledge that neuroimaging analysis of mild TBI exhibiting no gross pathology can typically be accommodated using standard algorithms, although for moderate and/or severe TBI more sophisticated methods are needed, as previously stated. Given the fact that most automatic segmentation algorithms have been developed for healthy brains or for brains with diminutive amounts of gross pathology (Irimia et al., 2012c), implementing such algorithms for moderate to severe TBI cases often necessitate periodic user intervention and guidance. This suggests that future data-processing workflows devised for facilitating TBI segmentation should aim to accommodate and minimize the need for periodic user intervention. Presently, a persistent challenge resides in the methodological dichotomy of opting for either a manual or automatic segmentation approach. While manual segmentation methods do not require (complex) segmentation algorithms, such methods are significantly more costly than automatic ones due to the comparably large amount of time and human resources needed for adequate segmentation of even a single MRI volume. Furthermore, the nature of manual delineation implies that substantial inter- and intra-observer variability are to be expected, which may increase quantitative measurement errors and thereby diminish the statistical power of inferential tests applied to sets of such measurements (Kempton et al., 2011). In the case of TBI lesions, however, one benefit of manual segmentation is that it is often more trustworthy than conventional automatic segmentation algorithms which were developed for the tissue classification of healthy brains (Lehmann et al., 2010), largely because TBI pathology is extremely heterogeneous among subjects. On the other hand, conventional automatic segmentation algorithms greatly reduce data processing time and improve reproducibility, but can suffer from appreciable inaccuracies in the case of TBI.

One software package which is often used for automatic segmentation and whose methodological capabilities are illustrative of automatic segmentation packages in general is FreeSurfer (Dale et al., 1999). As in the case of typical unsupervised segmentation packages, FreeSurfer has been thoroughly validated in healthy brains and in some diseases exhibiting structural pathology types which are more moderate and more predictable than those encountered in TBI (Du et al., 2007; Jovicich et al., 2009). Nevertheless, automatic tissue classification algorithms including FreeSurfer remain imperfect and can experience inaccuracies in skull stripping, WM/GM boundary identification, etc. even in healthy subjects (Strangman et al., 2010). Any such defects may require user input to (a) add control points and thereby aid FreeSurfer to identify WM, (b) remove unlabeled voxels representing the dura mater and thereby correct skull stripping, and/or (c) manually restore WM/GM portions which had been inappropriately removed during first-pass segmentation. In addition, typical automatic segmentation methods do not perform lesion classification, which suggests that additional user-guided segmentation is needed in this step as well.

In comparison to conventional tissue classification methods, a number of sophisticated segmentation methods now exist which have adopted more sophisticated approaches to address the task of TBI tissue segmentation. As stated, TBI pathology is often gross and highly heterogeneous, even in comparison to other types of neuropathology such as AD. In some cases, pathology patterns may present image intensities and appearance similar to those of normal tissues (Irimia et al., 2012c), necessitating segmentation algorithms tailored to analyzing pathology. In dealing with the similar problem of MR volume segmentation in multiple sclerosis (MS), Van Leemput et al. (2001) proposed a method which detects MS lesions as outliers with respect to a statistical model for the healthy brain, rather than attempting to model such lesions explicitly. The model interleaves (a) statistical classification of the image voxels into a number of healthy tissue types, (b) evaluation of whether each voxel truly belongs to healthy tissue, and (c) estimation of intensity distribution parameters and MR bias field parameters only based on healthy tissue voxels. Voxels not well constrained by the statistical model for normal brain MR images are detected as voxels containing MS lesions. Another sophisticated approach designed specifically for TBI (Irimia et al., 2012c; Wang et al., 2012) employs multimodal neuroimaging data from multiple time points to improve segmentations and to describe changes in healthy tissue and pathology. Their framework utilizes several semi-automatic segmentation tools available within 3D Slicer, a freely available software environment for image processing where automatic segmentation can be complemented by additional user evaluation (Irimia et al., 2011). Examples of semi-automatic segmentations obtained using such workflows are shown in **Figure 1**. Similar algorithms have been derived from approaches for the MR analysis of brain sclerosis and tumors, which present problems similar to those of TBI lesion segmentation (Prastawa et al., 2003, 2004). Other algorithms such as the one developed by Wu et al. (2006) use multimodal MRI to classify MS lesions into several subtypes, each of which can be analyzed to represent different outcome measurements.

Finally, because standard registration and segmentation methods do not account for changes in image appearance across time, sophisticated methods have been developed to jointly estimate a space deformation and a change in image appearance which can lead to the construction of a spatiotemporal trajectory which smoothly transforms the structural volume acquired from the patient at one time point into the volume acquired at a subsequent time point. In particular, algorithms such as that of Niethammer et al. (2011) have the ability to explain changes in image appearance by (a) a global deformation, (b) a deformation within a geometric model, and (c) an image composition model. The development of such longitudinal registration methods is motivated by the challenge to predict long-term effects of TBI based on longitudinal changes in tissue types and in their spatial configuration, which may provide further clinical insight into the prediction of tissue fate and patient outcome.

The wealth of MR segmentation algorithms is an indication that segmentation, at least in the case of TBI, is a complicated task which can be solved through many approaches. However, this wealth, arguably, is also an indication that no single approach has been demonstrably superior. Many of these methods, in fact, still require user intervention and post processing. Therefore, automatic segmentation may be an appropriate problem for the neuroinformatics community to address by means of data mining and novel workflow designs.

CONNECTIVITY NEUROIMAGING APPROACHES

As discussed in the previous section, conventional structural neuroimaging methods enable the calculations of volumetrics and morphometrics, which can reveal important information on gross anatomy changes effected by brain injury upon the brain in general and upon cortical structures in particular. By contrast, the advent of modern neuroimaging methods which allow the observation of neuronal circuitry *in vivo* (such as diffusion tensor imaging, DTI) has perpetuated the interest in connectivity mapping, and further allows investigation of connectivity changes in brain injury patients. The benefit of DTI in contrast to dissection and to WM staining is that the former can be used noninvasively in human patients, which is a major advantage in human studies. Techniques such as DTI tractography enable the mapping of macroscopic WM connections, which can yield descriptive metrics of brain connectivity, including fiber bundle length and connectivity density (Wang et al., 2012).

The ability of DTI tractography methods to reconstruct area-to-area connectivity in TBI has been the topic of multiple validation studies (Mori and van Zijl, 2002; Dauguet et al., 2007; MacDonald et al., 2007; Skudlarski et al., 2008), including one study by the present authors, where area-to-area connectivity counts obtained via DTI using purpose-built software were independently validated by three researchers with experience in

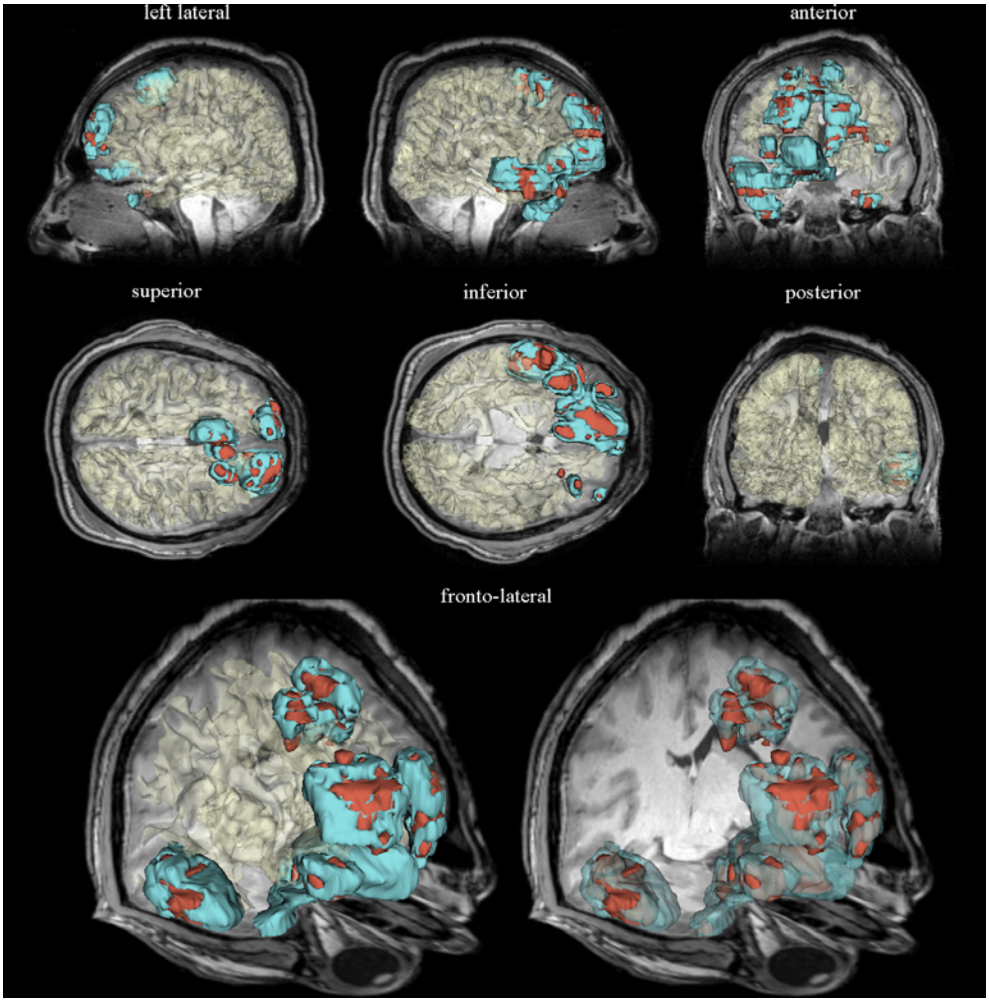


FIGURE 1 | Three-dimensional models of semi-automatically segmented healthy-appearing and pathology-affected tissues are displayed for a sample patient with severe TBI within a neuroinformatics framework. Representative slices of the T_1 volume acquired 3 days after injury are superimposed. Models of edematous and hemorrhagic tissues are colored in cyan and dark red, respectively. The WM surface was segmented automatically using FreeSurfer, demonstrating the capabilities of this software package to perform automatic tissue classification of healthy-appearing tissues. The WM model is translucent in each brain view to facilitate the visibility of anatomic details obviated in the MR volume slice displayed. See Irimia et al. (2011) for a detailed description of the neuroinformatics methodology used to generate these visualizations.

neuroanatomy (Van Horn et al., 2012). Whereas DTI is certainly not as accurate for reconstructing area-to-area connectivity as some invasive methods (e.g., *post-mortem* dissection and WM staining), its ability to capture connectivity information accurately has been found to be quite reasonable provided that the size of each brain parcel denoting a graph node is sufficiently large compared to the DTI voxel size (Irimia et al., 2011; Irimia et al., 2012c; Van Horn et al., 2012).

It has been acknowledged (Meythaler et al., 2001) that 40–50% of TBI patients exhibit diffuse axonal injury (DAI), a mechanism of brain injury which is microscopic in nature such that conventional CT and MRI are typically insufficient to capture it in detail. DTI, on the other hand, is more ideally suited to non-invasively measure the diffusion of molecules through biological tissue. Whereas diffusion of water along healthy axons is predominantly anisotropic, studies using DTI have indicated that DAI may be detected as a reduction in diffusion anisotropy (Arfanakis et al., 2002). With the advancement of such techniques, the goal of characterizing TBI-related changes in brain connectivity can be pursued by using brain water diffusion data to reconstruct WM tracts three-dimensionally, to visualize fiber cluster integrity and to locate gross anatomy changes prompted by injury.

To study WM changes prompted by TBI, neuroimaging researchers have adopted various mathematical approaches to aid in data analysis, the most prominent of these being network theory. This approach typically focuses upon the task of reconstructing brain networks using graphs, which are mathematical representations consisting of nodes (vertices) and links (edges) between pairs of nodes. Such representations have long been used to represent brain networks (Strogatz, 2001), though their popularity for the purpose of systematic connectivity mapping in humans via non-invasive techniques such as DTI has only increased appreciably throughout the past decade (Bullmore and Sporns, 2009; Rubinov and Sporns, 2010). In the context of neural connectivity, nodes represent brain regions which exhibit some given functional or anatomical pattern. Links, on the other hand, denote the presence or absence of connections, and can be weighted to represent the strengths of neural connections between distinct areas (Strogatz, 2001; Rubinov and Sporns, 2010). The manner in which nodes and links are defined can vary substantially, depending on the set of conventions used to parcellate the brain. In many cases, parcellation schemes are used to delineate gyri and sulci into homogenous regions which correspond to graph nodes (Thirion et al., 2010; Stanley et al., 2013). The advantage of this approach is that each graph node corresponds to an anatomical region whose identity and spatial extent have been well documented by neuroanatomists (Irimia et al., 2012c; Van Horn et al., 2012).

The application of network theory within TBI neuroinformatics has increased in recent years (Achard et al., 2012; Irimia et al., 2012c; Van Horn et al., 2012; Wang et al., 2012), with a special focus upon identifying network patterns which can offer insight into the long-term effects of TBI. A study by Pandit et al. (2013), for example, utilizes the tools of network theory to investigate changes in brain network topology following TBI, to the effect that the victims of this condition exhibit abnormalities with respect to normal controls from the standpoint of several global network-theoretic measures, including total connectivity, average path length and network efficiency. Thus, one advantage of DTI which is highly beneficial to the study of TBI is the fact that this imaging modality allows the extraction of network-theoretic connectivity information from which patient-specific measures can be computed, including metrics of centrality, assortativity, node degree, etc. (Achard et al., 2012; Irimia et al., 2012a). Statistical comparison of such measures between TBI patients and healthy control subjects can outline the nature, extent and location of TBI damage upon neural pathways, and may also reveal information which can be useful when formulating personalized rehabilitation strategies.

Network metrics can be used to investigate patterns of connectivity changes in TBI patients and to inform clinicians who wish to incorporate the use of this knowledge into the process of treatment formulation. This trend is already under way in the study of other disorders of the nervous system; for example, previous studies have found significant differences in network-theoretic metrics (e.g., spatial pairwise clustering and intra-nodal homogeneity) when comparing healthy adults to schizophrenics (Zalesky et al., 2012), AD patients, and to normal aging. Thus, the informatics relevant to these studies offers new ways to quantitatively characterize changes in anatomical network patterns, including the means to relate WM network topology to brain function. These techniques are particularly relevant in TBI due to the well-known facts that (a) brain injury can cause dramatic changes in WM connectivity (Kinnunen et al., 2011; Irimia et al., 2012a) and that (b) such changes often result in the deterioration of cognitive function (McDowell et al., 1997; Chen and D'Esposito, 2010). Because cognitive deficits incurred as a result of injury may either ameliorate or deteriorate over time depending on a variety of factors (Hoofien et al., 2001; Kraus et al., 2007), neuroinformatics approaches designed for professionals in the field of TBI (e.g., TBI clinicians, epidemiologists, public health professionals, etc.) are well-suited for providing clinicians and researchers with advanced tools for investigating the temporal evolution of TBI WM lesion profiles. This may lead to an improvement of current understanding on how neurological damage leads to functional impairment, and may also spur the development of pathology-tolerant neuroimage analysis tools which can be applied to other types of brain injury, such as stroke and MS.

Despite the widespread application of diffusion imaging over the years, several fundamental technical challenges remain only partially resolved. One persistent difficulty has been the challenge of correcting for head movement in the MR scanner. Head motion not only interferes with image acquisition, but may also lead to errors in the calculation of diffusion tensor scalars such as fractional anisotropy (FA) and mean diffusivity (MD), as shown in a number of studies (Ling et al., 2012; Van Dijk et al., 2012). It should be noted that head motion is not unique to connectivity neuroimaging and that it is also a concern in structural neuroimaging. Approaches to mitigating head motion in non-head injury patients have included the use of anesthesia (Karlik et al., 1988; Holshouser et al., 1993), which is often used when neuroimaging data are acquired from acute injury patients in a neurointensive care setting. Naturally, however, this approach may not be suitable in all TBI cases, and therefore the integration of motion correction algorithms into post-processing steps remains critical to the usability of the acquired data. Investigators have systematically examined the residual effects of head motion in diffusion imaging, and have reported the impact of head motion upon the calculation of diffusion metrics. Tijssen et al. (2009) found a positive bias between head motion and FA in regions with low anisotropy; in regions with higher anisotropy, head motion was found by these authors to artifactually decrease FA. Ling et al. (2012) reproduced these findings and expanded on the findings of Tijssen et al. by examining the residual effects of motion following conventional motion correction frameworks (i.e., image registration, gradient table adjustment, diffusion weighted image removal). This is especially problematic in TBI studies where diffusion metrics may incorrectly represent the presence or absence of pathology-affected tissue. Thus, further research into the development of effective motion correction algorithms is particularly critical in the context of TBI research.

Another challenge resides in the somewhat limited ability of tracking algorithms to correctly infer the continuity of fibers from voxel to voxel. One drawback of probabilistic tractography which can affect TBI studies with predilection is that the latter is more likely to reconstructs short fibers, which can increase the probability that WM located near GM or near a lesion is assigned an inappropriately large number of tracts (Kuceyeski et al., 2011). Connectivity assessment may further be complicated by the presence of edematous or hemorrhaging tissue, where the appreciable isotropy of water diffusion interferes with the ability of DTI to capture fiber directionality. Yet another factor which TBI neuroinformatics tools should aim to account for is the difficulty of detecting crossing fiber bundles, particularly in peri-lesional regions. This phenomenon, which is traditionally known to be caused by limitations in current approaches for reconstructing fiber trajectories (Iturria-Medina et al., 2007; Bullmore and Sporns, 2009), can be particularly challenging to account for in TBI where changes in anisotropy are often prompted within and surrounding lesion sites.

A final concern for TBI connectivity analysis is the increasing need for versatile data visualization tools. While a large number of these exist, many of them, as Margulies et al. (2013) point out, are limited by the necessity to compromise and prioritize the representation of information in terms of anatomic vs. connectomic, aesthetics vs. informational content, and thoroughness vs. readability. For example, although *TrackVis* is intended for whole brain tractography visualization, its strength is primarily in data visualization rather than data processing and computation. By comparison, *OpenWalnut* is more tailored towards data processing due to the modularity of its software environment design and pipelining engine. However, while both of these tools fulfill the need for anatomic visualization, very few workflows exist which offer comprehensive summaries (i.e., anatomy, function, connectivity) of the human connectome as reconstructed via neuroimaging.

Research on mapping and visualizing cortical connections has a relatively long history, beginning with animal model studies. Scannell and Young, in particular, have performed extensive work on the cat cerebral cortex in representing neural connectivity using a variety of graph depiction strategies (Scannell and Young, 1993; Scannell et al., 1995). Irimia et al. (2012a) have developed a graphical approach for representing TBI connectivity alterations, illustrating the location and extent of WM change over time in TBI patients. This visualization paradigm generates connectivity representations called “connectograms” using an informatically driven software package which allows brain connectivity information to be depicted within a circle of radially aligned elements. A connectogram from a sample TBI patient created using this approach is shown in **Figure 2**. The purpose of this figure is to illustrate the presence of appreciable atrophy due to TBI. Each circular wedge element represents a specific cortical region and is positioned on either side of the vertical axis, corresponding to the left or right hemisphere, respectively. The location of each fiber extremity is associated with the appropriate cortical parcellation of a sulcus or gyrus. Inter-region connectivity is represented by a link of variable opacity drawn the between radially aligned elements, and depends on fiber density as well as upon pathology severity. This mode of representation emphasizes the presence of atrophy, which is substantially more severe in TBI than in healthy aging, particularly over a 6-month period. For this reason, in contrast to **Figure 2**, it is to be expected that the connectogram displaying longitudinal changes in connectivity for a healthy adult would reveal considerably fewer and weaker changes over a 6-month period, particularly for a young or middle-aged adult.

The connectogram as a graphical representation method offers a succinct means of displaying longitudinal differences in WM connections and highlights the current impetus for incorporating neuroinformatics approaches into the development of brain connectivity visualization methods (Margulies et al., 2013). Advances in robust connectivity visualization and representation methods could encourage longitudinal studies, which depend on neuroinformatically driven workflows to process the large amounts of data associated with capturing and quantifying connectivity changes across multiple time points. Armed with measurements of morphologic and connectomic alterations over time, customized publication database search strings may additionally be crafted and submitted to PubMed or Google Scholar to return literature relevant to damage in the affected areas, the effects on connectivity, and putative treatment options (Irimia et al., 2012a). Recent approaches to information retrieval, extraction and analysis of the neuroimaging literature, such as those of Bug et al. (2008) and Keator et al. (2013) may provide additional starting points for the development of flexible tools for the description and retrieval of neuroscience-relevant resources, as pioneered by the Neuroscience Information Framework (NIF).

FUNCTIONAL IMAGING AND NEUROPHYSIOLOGICAL APPROACHES

Functional neuroimaging modalities and electrophysiological recordings allow researchers to investigate behavioral deficits as well as the pathophysiological responses of the brain following injury. The techniques most frequently employed include functional MRI (fMRI), electroencephalography (EEG), magnetoencephalography (MEG), and PET. Each of these techniques possesses varying levels of applicability with inherent strengths and weaknesses depending on the aims of the study, as well as on the condition of the patient. Accordingly, it would be beneficial to develop data mining, processing and analysis approaches which can facilitate the optimization of information usage acquired across various functional imaging modalities.

Whereas fMRI is useful in post-injury investigations of cerebral activation patterns during the performance of cognitive tasks, its reliability in diagnostic applications may be impeded by factors such as increased intracranial pressure, which can alter hemodynamic responses and, subsequently, its measure of cerebral activity (Hillary et al., 2002). In such cases, the use of EEG may be preferable to that of fMRI or PET due to the high temporal resolution of the former (in the millisecond range), and to the fact that EEG does not rely on indirect measures of activity such as the hemodynamic response. Nevertheless, it is useful to note that the temporal resolution gap between fMRI and EEG may be partially alleviated through the use of novel multi-band methods for fMRI, which involve shorter acquisition times and thus greater temporal resolution (Moeller et al., 2010; Ugurbil, 2012). One limitation of EEG to consider, however, is the fact that the structural changes and presence of pathology prompted by TBI may increase the difficulty of localizing pathophysiological activity recorded after acute brain injury. Specifically, electrical source localization is a problematic task due to the ill-posed nature of the bioelectric inverse problem. The latter refers to the task of localizing the sources of brain activity based on scalp EEG measurements. By contrast, the calculation of electric potentials produced at the scalp due to current sources in the brain is known as the forward problem of bioelectricity (Lima et al., 2006; Irimia et al., 2013a). Additionally, appreciable cancellation of cortical signals occurs in EEG (Goh et al., 2013; Irimia et al., 2013a,b). Accurate localization of cortical activity depends on a number of factors, one of which is the

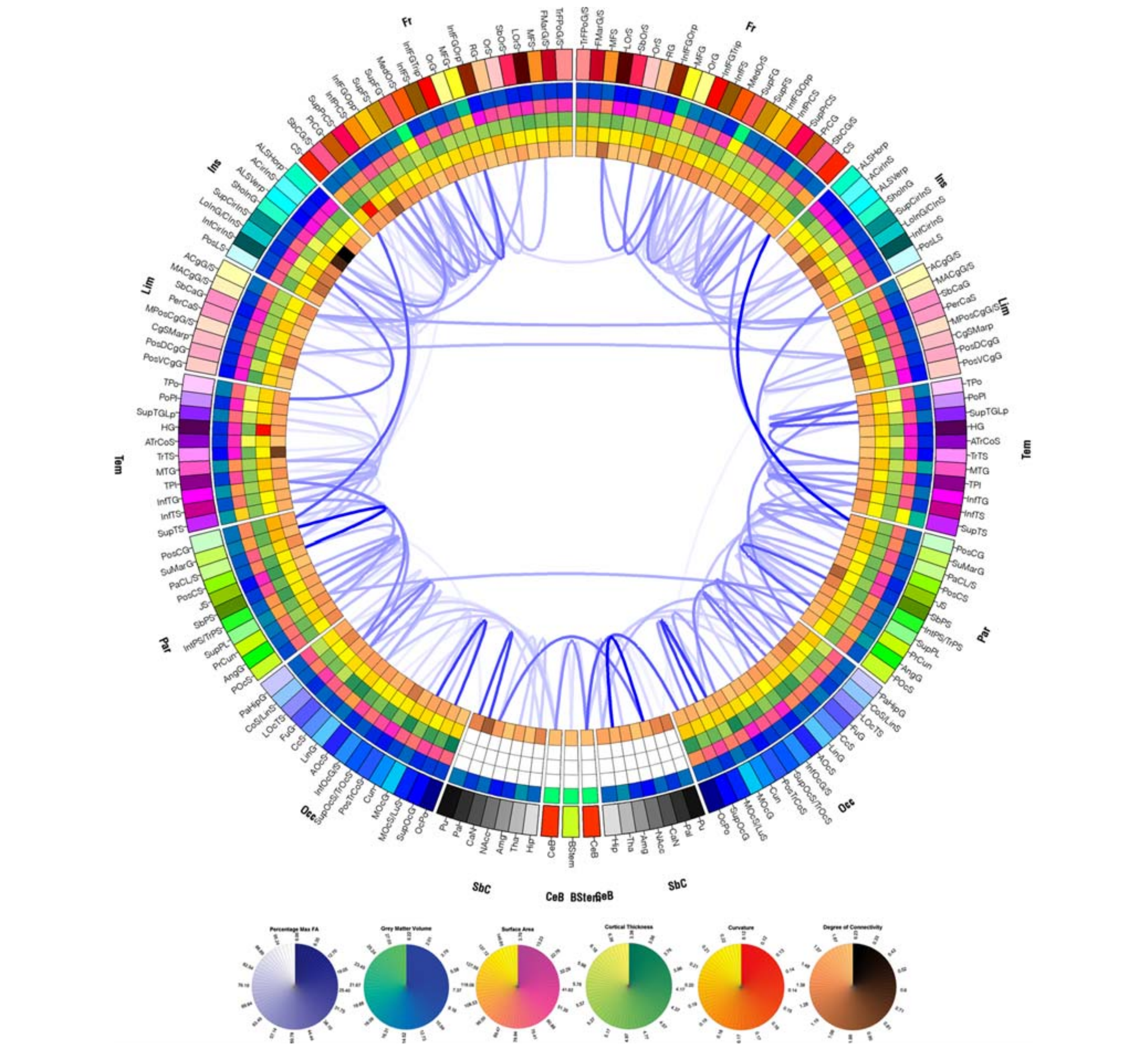


FIGURE 2 | Circular connectogram representation graphically displays WM atrophy over a 6 month period. The left and right halves of the connectogram correspond to the left and right hemispheres, respectively. Each hemisphere of the brain is divided into frontal, insular, limbic, temporal, parietal and occipital lobes, as well as into subcortical structures, cerebellum, and the brain stem; the latter three are represented at the bottom of the circle. Each lobe is further divided into parcels (gyri and sulci in the case of the cortex) and is assigned a unique identifying color. Radially aligned, concentric rings represented using various color schemes depict various attributes of each corresponding brain parcel. From the outermost to the innermost one, the rings contain wedges which encode GM volume, surface area, cortical thickness, curvature, and degree of connectivity. A link of variable opacity is drawn between certain pairs of brain parcels, reflecting structural connectivity properties between regions. In the case of the connectogram displayed, links displayed indicate connections which suffered from large atrophy from the acute baseline to the chronic follow-up time point. Link transparency encodes the percentage change Δ in fiber density, in the range [min(Δ), max(Δ)], with larger changes (more negative values of Δ) being encoded by more opaque hues of blue. The lowest color opacity corresponds to the smallest absolute value of the percentage change which is greater than the selected threshold of 30%, and the highest opacity corresponds to the maximum absolute value of the change in fiber density. See Irimia et al. (2012a) for details.

anatomic faithfulness of the head model used in the forward calculation of electric potentials (Gencer and Acar, 2004; Goh et al., 2013). EEG localization studies involving models which account for the presence of lesions and cavities have shown that the latter can have significant qualitative and quantitative effects upon the computed electric potentials (He et al., 1987). Thus, from an informatics standpoint, it is necessary to develop data processing tools which incorporate realistic head model generation and which can account not only for head anatomy and tissue conductivity profiles, but also for the effects of tissue conductivity changes due to TBI. Comparatively, MEG presents advantages which are unique and often complementary to EEG. For example, a single head volume model is typically sufficient in MEG forward modeling, partly because the biomagnetic fields of the brain are far more dependent on tissue permeability rather than conductivity. Whereas conductivity can vary considerably across biological tissues, their permeability is always very nearly equal to that of free space (μ_0), such that the use of a single head volume is justified. An immediate consequence of this fact is that, whereas the spatial distribution of electric potentials over the scalp is smeared and attenuated due to the high resistivity of the skull, magnetic field recordings are nowhere near as strongly affected by the conductivity profile of the head, which is advantageous in MEG experiments (Lima et al., 2006; Sharon et al., 2009; Irimia et al., 2012b). In addition, the number of sensors used for MEG recordings (e.g., 306 sensors in the Elekta Neuromag® MEG scanner) is often higher than that of EEG montages, where fewer than 256 sensors are typically used. Finally, MEG sensors can usually sample brain signals at higher frequencies and signal-to-noise ratios than EEG electrodes. Nonetheless, MEG scanners are available only at a handful of brain research centers, and data acquisition costs for this modality are prohibitively higher than for EEG. Future data-processing tools devised for acquiring and analyzing brain signals from TBI patients should aim to be user-friendly, regardless of whether EEG or MEG is used. In this context, the requirement of user friendliness implies that the approaches for data acquisition and analysis should be intuitive to grasp and easy to use by clinicians and by other health professionals who are unfamiliar with the complexities of anatomical modeling and of inverse localization methods for EEG-based neurophysiological signal analysis.

Although a variety of functional neuroimaging and electrophysiological techniques can and have been used in neurotrauma research, a large number of functional TBI studies are uni-modal in the sense that they employ only a single technique to obtain quantitative values of a specific measure. Naturally, it would be more advantageous to combine multiple modalities in order to achieve a more comprehensive view of how brain injury leads to subsequent functional losses. An insufficient number of studies have accomplished this, however, due to the difficulty associated with integrating data acquired across various measurement modalities. Research involving the localization of brain activity after TBI using EEG includes three recent studies (Goh et al., 2013; Irimia et al., 2013a,b) where the combined use of MRI and EEG is demonstrated. In both of these studies, cortical electrical activity is inversely mapped over the cortex with clinical applications to the localization of epileptogenic foci in post-traumatic epilepsy (PTE). An example of this approach is shown in **Figure 3**. In these studies, the effects of pathology upon forward modeling and inverse source localization were explored in the context of a semi-automatic, multimodal neuroimaging approach involving anatomically faithful TBI head models containing 25 tissues types, including six types accounting for TBI-related pathology. The multimodal aspects of these studies highlight the combined use of structural and functional imaging data using an inverse localization algorithm subject to anatomic constraints provided by MRI.

In a general sense, neuroimaging-based methodologies have not yet addressed the paucity of strategies for integrating multivariate connectivity data with other imaging modalities including fMRI, PET, EEG, and MEG. The ability to extract meaningful information from multimodal data must often make use of dimensionality reduction techniques, as well as multivariate statistical inference methods which can allow researchers to test statistical hypotheses based on large descriptive feature vectors. One study which illustrates the integration of functional neuroimaging modalities to the benefit of TBI research is by Storti et al. (2012), who integrated fMRI and EEG to evaluate PTE in patients with pharmacologically resistant epilepsy. During MRI scanning, the patients who participated in this study were additionally equipped with an MR-compatible EEG amplifier and cap arranged in the 10/20 montage. The combined use of these modalities allowed the authors to compare clinical semiology, BOLD activation, and source localization which could only be obtained as a result of

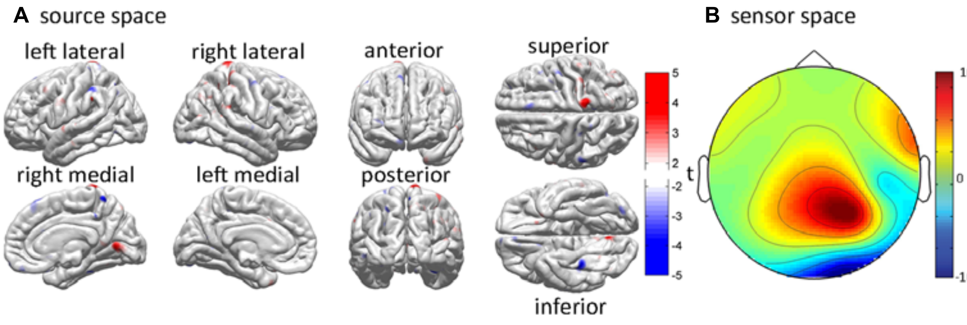


FIGURE 3 | Example of EEG inverse localization in a sample acute TBI patient using an integrative pipeline. The cortical sources responsible for the generation of recorded EEG waveforms are determined using the application of a minimum norm inverse localization method. **(A)** EEG potentials recorded over the scalp (i.e., in "sensor space") are inversely localized onto the cortical surface (i.e., into "source space"). The inverse estimate of the cortical activity responsible for the generation of EEG signals is plotted using *t* scores, which indicate the likelihood for each cortical location to be electrically active. The magnitude of *t* indicates whether the localized electric current is oriented out of ($t > 0$, red hues) or into ($t < 0$, blue hues) the cortex. **(B)** The interpolated values of the potentials measured at each sensor location are mapped over an idealized, circular representation of the scalp to generate a topographic map. Color indicates the magnitude of the recorded electric potential Φ in μV. See Irimia et al. (2013b) for further details.

the advantages offered by the complementary nature of combined fMRI/EEG. As previously stated, conventional fMRI alone offers high spatial resolution, but poor temporal resolution, whereas EEG alone offers high temporal resolution but relatively poor spatial resolution in the absence of inverse localization. Multimodal neuroimaging is ideally suited for TBI clinical care because different modalities can reveal distinct information about injury. For example, an MRI FLAIR sequence can reveal the presence and spatial extent of brain edema, whereas an SWI sequence is ideally suited for the detection of microhemorrages. Thus, the fusion of such multimodal information can provide substantial insight into the structural profiles of lesions, thereby helping to formulate clinical interventions. Nevertheless, despite the trend toward integration of modalities to study TBI across all its stages, it has been proposed that the use of fMRI and PET is more appropriate during the sub-acute to chronic stages, as opposed to the acute phase where the presence of increased intracranial pressure is likely and may lead to misleading measurements (Hillary et al., 2002). In chronic TBI, by contrast, metrics of brain function derived from fMRI and PET have been used by various researchers to investigate neuropsychiatric performance (Kasahara et al., 2011; Palacios et al., 2013).

The motivation for diversifying the range of functional neuroimaging modalities which are typically included in analyses of brain structure has increased considerably as neuroimaging analysis methods have become more sophisticated. In this respect, one key point to address in functional TBI neuroimaging studies is the fact that large volumes of data are often generated in the course of neuroimage acquisition and analysis. Specifically, data acquired using modalities such as fMRI, EEG and MEG incorporate a time dimension: (a) in the case of multiband fMRI, the additional 3D nature of this modality can make data storage a very substantial challenge; (b) in the cases of EEG and MEG, the high temporal resolution (in the MHz range, though typically down-sampled to the kHz range or lower) can also raise storage-related challenges. Collectively, these properties of functional neuroimaging data can result in substantial storage demands from dedicated databases and repositories (Van Horn and Toga, 2009). An examination of fMRI articles from representative issues of the journal *Neuroimage* found that since 1995, the amount of data collected has doubled approximately every 26 months (Van Horn and Toga, 2009, 2013). At this rate, it is projected that data storage requirements may exceed 20 GB per published study by the year 2015. Consequently, it is vital that funding agencies should support the computational infrastructure needed to accommodate multimodal data, and that hardware resource availability should develop alongside at the same pace. Next-generation neuroinformatics approaches to the management of multimodal data should also be developed, particularly for the purpose of inter-institutional collaborations and data sharing.

An important recent trend in the consideration of functional TBI neuroimaging has been the proliferation of approaches involving data-intensive discovery – rather than hypothesis testing – in TBI research (Akil et al., 2011). The net result of this trend has been the need for centralized databases to assist the research community in terms of hardware infrastructure and efficiency of data mining. Whereas a number of neuroimaging databases exist which are dedicated to the gathering and dissemination of neuroimaging data for various types of diseases including ADNI (Jack et al., 2008; Jack et al., 2010; Weiner et al., 2012), such large-scale database systems are only now becoming available for the purpose of TBI neuroimaging research, including the informatics system of the Federal Interagency Traumatic Brain Injury Research (FITBIR, fitbir.nih.gov). In addition to FITBIR, the NIF (www.neuroinfo.org) is another useful resource established to survey and compile a list of neuroscience databases, tools, and materials so that researchers can efficiently search across a variety of smaller, individual databases.

For FITBIR, NIF and other resources and databases dedicated to the task of disseminating data and functional neuroimaging analysis software to the research community, one challenge which requires careful consideration is the need for data sharing and storage mechanisms to accommodate large collaborations across multiple research centers with wide geographic distributions. The intrinsic necessity for multidimensionality in TBI neuroimaging data sets entails the reality that inter-institutional TBI research may require hardware data storage capabilities in excess of those needed by other large neuroimaging collaborative efforts such as ADNI, for example, which does not need to rely as heavily as TBI research does upon data multimodality. Furthermore, it would be highly beneficial for researchers to benefit from neuroinformatics-driven data sharing capabilities which can facilitate collaborations among researchers from various institutions as well as among clinical and research staff responsible for acquiring TBI neuroimaging data (Manley and Maas, 2013).

DISCUSSION

Despite the emerging trend towards the use of multimodal imaging by TBI experts, the capacity to acquire and process large amounts of neuroimaging data remains dependent upon the availability of sophisticated imaging hardware and large-scale computational resources to store and manage such data. Additionally, extracting meaningful and clinically useful information from multimodal neuroimaging data can necessitate advanced neuroimaging processing software packages which are capable of handling their multi-dimensionality and inherent complexity. Although improvement of TBI treatment and rehabilitation protocols by means of multimodal neuroimaging remains a critical goal to healthcare providers, much of the ability to accomplish this aim is dependent upon the identification of clinical biomarkers which are predictive of TBI pathology progression, and the future of TBI neuroinformatics must therefore accommodate the use of statistical prediction models which aid in forecasting TBI clinical outcome.

Computational neuroanatomy can aid TBI outcome prediction by providing quantitative metrics for further analysis rather than by resorting to the task of discerning voxel intensity differences visually or to similar types of qualitative observations. By definition, quantitative structural imaging studies utilize mathematical computations which can be reliably reproduced and applied across entire cohorts, and such undertakings can be facilitated through the use of neuroinformatics. Nevertheless, when considering the task of performing inferential statistical analyses of neuroimaging-derived structural metrics in TBI, it is also critical to incorporate statistical techniques which can accommodate and account for the attrition rates encountered in longitudinal studies of this condition. Specifically, one-third to one-half of TBI study participants are lost to follow-up primarily due to low socioeconomic status, substance abuse history, and violent injury etiology (Corrigan et al., 2003). This can be detrimental to the validity of outcome studies, and data processing workflows tailored for structural neuroimaging analyses should therefore implement biostatistical techniques for addressing the problem of missing measurement data in order to account for the attrition rates encountered in longitudinal studies of this population.

The wealth of information which can be extracted from connectivity analyses has spurred the development of graph-theoretic quantitative approaches to describe brain network organization following TBI. The methodologies of classical graph theory have lent their power to the study of complex networks such as those in the brain, and the resulting approaches have been beneficial to the task of quantifying the networks of the brain with high reliability and reproducibility using a manageable number of neurobiologically meaningful and easily computable quantitative measures (Rubinov and Sporns, 2010). Furthermore, network-theoretic metrics can be robust to the use of distinct cortical parcellations across studies as well as to various approaches for quantifying functional connectivity. This is particularly useful in the case of TBI because investigating relationships between brain structure, neurological damage, and functional impairment is essential when attempting to formulate patient-specific rehabilitation protocols.

The goals of numerous TBI neuroimaging studies can be greatly facilitated by the use of neuroinformatics protocols to streamline and perform data analysis, but the availability solutions to facilitate the study of brain structure, function and connectivity remains insufficient. This is partly due to the intricate complexities of the human brain and its functions, and partly due to the fact that neuroimaging-based methodologies for its study have not yet fully matured. Structural, connectomic, and functional data are highly multidimensional, which frequently demands the use of sophisticated statistical methods for multivariate analysis. Current data processing efforts for their joint analysis continue to be hampered by the need for considerable manual customization steps which are often needed to bridge compatibility gaps between the various software environments employed. For instance, to perform anatomically faithful forward/inverse calculations in EEG, head model generation requires not only the segmentation of healthy-appearing tissues – which can be performed more or less automatically – but also the segmentation of pathology-affected tissues, which is often performed manually, as outlined in the first section. However, because little compatibility typically exists across software environments and the algorithms used for each of these processing steps, neuroinformatically informed strategies are necessary to invoke the integration of neuroimage segmentation tools with forward model generation modules, inverse localization algorithms, and other methodologies for the analysis of brain functional data.

In conclusion, next-generation TBI neuroinformatics must address the need to develop integrative workflows which (a) perform automatic tissue segmentation of TBI pathology, (b) lead to a reduction in the number of algorithmic approaches and software environments required for connectomic and functional analysis, (c) minimize the amount of time and effort devoted by the user to manual intervention, and which (d) promote knowledge extraction leading to targeted clinical intervention. Such integration can allow researchers to generate strategies for analyzing brain function after injury, for extracting clinically useful information from each modality, for combining information obtained from each modality, and for gaining insight into the relationships between brain metabolism, cerebral blood flow, and cortical electrical activity underlying successful recovery in TBI.

ACKNOWLEDGMENTS

The authors wish to thank the faculty and staff of the Institute for Neuroimaging and Informatics at the University of Southern California. This work was supported by NIH grants R41 NS081792 and U54 EB005149, sub-awards to John D. Van Horn.

REFERENCES

Achard, S., Delon-Martin, C., Vertes, P. E., Renard, F., Schenck, M., Schneider, F., et al. (2012). Hubs of brain functional networks are radically reorganized in comatose patients. *Proc. Natl. Acad. Sci. U.S.A.* 109, 20608–20613. doi: 10.1073/pnas.1208933109

Akil, H., Martone, M. E., and Van Essen, D. C. (2011). Challenges and opportunities in mining neuroscience data. *Science* 331, 708–712. doi: 10.1126/science.1199305

Arfanakis, K., Haughton, V. M., Carew, J. D., Rogers, B. P., Dempsey, R. J., and Meyerand, M. E. (2002). Diffusion tensor MR imaging in diffuse axonal injury. *Am. J. Neuroradiol.* 23, 794–802.

Ashburner, J., and Friston, K. J. (2000). Voxel-based morphometry–the methods. *Neuroimage* 11, 805–821. doi: 10.1006/nimg.2000.0582

Bug, W. J., Ascoli, G. A., Grethe, J. S., Gupta, A., Fennema-Notestine, C., Laird, A. R., et al. (2008). The NIFSTD and BIRNLex vocabularies: building comprehensive ontologies for neuroscience. *Neuroinformatics* 6, 175–194. doi: 10.1007/s12021-008-9032-z

Bullmore, E. T., and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and fucntional systems. *Nat. Rev. Neurosci.* 10, 186–198. doi: 10.1038/nrn2575

Chen, A. J., and D'Esposito, M. (2010). Traumatic brain injury: from bench to bedside to society. *Neuron* 66, 11–14. doi: 10.1016/j.neuron.2010.04.004

Corrigan, J. D., Harrison-Felix, C., Bogner, J., Dijkers, M., Terrill, M. S., and Whiteneck, G. (2003). Systematic bias in traumatic brain injury outcome studies because of loss to follow-up. *Arch. Phys. Med. Rehabil.* 84, 153–160. doi: 10.1053/apmr.2003.50093

Dale, A. M., Fischl, B., and Sereno, M. I. (1999). Cortical surface-based analysis–I. Segmentation and surface reconstruction. *Neuroimage* 9, 179–194. doi: 10.1006/nimg.1998.0395

Dauguet, J., Peled, S., Berezovskii, V., Delzescaux, T., Warfield, S. K., Born, R., et al. (2007). Comparison of fiber tracts derived from in-vivo DTI tractography with 3D histological neural tract tracer reconstruction on a macaque brain. *Neuroimage* 37, 530–538. doi: 10.1016/j.neuroimage.2007.04.067

Du, A. T., Schuff, N., Kramer, J. H., Rosen, H. J., Gorno-Tempini, M. L., Rankin, K., et al. (2007). Different regional patterns of cortical thinning in Alzheimer's disease and frontotemporal dementia. *Brain* 130, 1159–1166. doi: 10.1093/brain/awm016

Faul, M., Xu, L., Wald, M. M., and Coronado, V. G. (2010). *Traumatic Brain Injury in the United States: Emergency Department Visits, Hospitalizations and Deaths 2002–2006.* Atlanta, GA: Centers for Disease Control and Prevention, National Center for Injury Prevention and Control.

Filippi, M., Gawne-Cain, M. L., Gasperini, C., Vanwaesberghe, J. H., Grimaud, J., Barkhof, F., et al. (1998). Effect of training and different measurement strategies on the reproducibility of brain MRI lesion load measurements in multiple sclerosis. *Neurology* 50, 238–244. doi: 10.1212/WNL.50.1.238

Gencer, N. G., and Acar, C. E. (2004). Sensitivity of EEG and MEG measurements to tissue conductivity. *Phys. Med. Biol.* 49, 701–717. doi: 10.1088/0031-9155/49/5/004

Ghajar, J. (2000). Traumatic brain injury. *Lancet* 356, 923–929. doi: 10.1016/S0140-6736(00)02689-1

Goh, S. Y. M., Irimia, A., Torgerson, C. M., Kikinis, R., Vespa, P. M., and Van Horn, J. D. (2013). "High-resolution electroencephalographic forward modeling in traumatic brain injury using the finite element method," in *2013 IEEE 10th International Symposium on Biomedical Imaging*, San Francisco, CA, USA.

He, B., Musha, T., Okamoto, Y., Homma, S., Nakajima, Y., and Sato, T. (1987). Electric dipole tracing in the brain by means of the boundary element method and its accuracy. *IEEE Trans. Biomed. Eng.* 34, 406–414. doi: 10.1109/TBME.1987.326056

Hillary, F. G., Steffener, J., Biswal, B. B., Lange, G., Deluca, J., and Ashburner, J. (2002). Functional magnetic resonance imaging technology and traumatic brain injury rehabilitation: guidelines for methodological and conceptual pitfalls. *J. Head Trauma Rehabil.* 17, 411–430. doi: 10.1097/00001199-200210000-00004

Hofmann, M., Bezrukov, I., Mantlik, F., Aschoff, P., Steinke, F., Beyer, T., et al. (2011). MRI-based attenuation correction for whole-body PET/MRI: quantitative evaluation of segmentation- and atlas-based methods. *J. Nucl. Med.* 52, 1392–1399. doi: 10.2967/jnumed.110.078949

Holshouser, B. A., Hinshaw, D. B. Jr., and Shellock, F. G. (1993). Sedation, anesthesia, and physiologic monitoring during MR imaging: evaluation of procedures and equipment. *J. Magn. Reson. Imaging* 3, 553–558. doi: 10.1002/jmri.1880030320

Hoofien, D., Gilboa, A., Vakil, E., and Donovick, P. J. (2001). Traumatic brain injury (TBI) 10-20 years later: a comprehensive outcome study of psychiatric symptomatology, cognitive abilities and psychosocial functioning. *Brain Inj.* 15, 189–209. doi: 10.1080/026990501300005659

Irimia, A., Chambers, M. C., Alger, J. R., Filippou, M., Prastawa, M. W., Wang, B., et al. (2011). Comparison of acute and chronic traumatic brain injury using semi-automatic multimodal segmentation of MR volumes. *J. Neurotrauma* 28, 2287–2306. doi: 10.1089/neu.2011.1920

Irimia, A., Chambers, M. C., Torgerson, C. M., Filippou, M., Hovda, D. A., Alger, J. R., et al. (2012a). Patient-tailored connectomics visualization for the assessment of white matter atrophy in traumatic brain injury. *Front. Neurol.* 3:10. doi: 10.3389/fneur.2012.00010

Irimia, A., Van Horn, J. D., and Halgren, E. (2012b). Source cancellation profiles of electroencephalography and magnetoencephalography. *Neuroimage* 59, 2464–2474. doi: 10.1016/j.neuroimage.2011.08.104

Irimia, A., Wang, B., Aylward, S. R., Prastawa, M., Pace, D. F., Gerig, G., et al. (2012c). Neuroimaging of structural pathology and connectomics in traumatic brain injury: toward personalized outcome prediction. *Neuroimage Clin.* 1, 1–17. doi: 10.1016/j.nicl.2012.08.002

Irimia, A., Goh, S. Y., Torgerson, C. M., Chambers, M. C., Kikinis, R., and Van Horn, J. D. (2013a). Forward and inverse electroencephalographic modeling in health and in acute traumatic brain injury. *Clin. Neurophysiol.* 124, 2129–2145. doi: 10.1016/j.clinph.2013.04.336

Irimia, A., Goh, S. Y. M., Torgerson, C. M., Stein, N. R., Chambers, M. C., Vespa, P. M., et al. (2013b). Electroencephalographic inverse localization of brain activity in acute traumatic brain injury as a guide to surgery, monitoring and treatment. *Clin. Neurol. Neurosurg.* 115, 2159–2165. doi: 10.1016/j.clineuro.2013.08.003

Iturria-Medina, Y., Canales-Rodríguez, E. J., Melie-García, L., Valdés-Hernández, P. A., Martínez-Montes, E., Alemán-Gómez, Y., et al. (2007). Characterizing brain anatomical connections using diffusion weighted MRI and graph theory. *Neuroimage* 36, 645–660. doi: 10.1016/j.neuroimage.2007.02.012

Jack, C. R., Bernstein, M. A., Fox, N. C., Thompson, P. M., Alexander, G., Harvey, D., et al. (2008). The Alzheimer's Disease Neuroimaging Initiative (ADNI): the MR imaging protocol. *J. Magn. Reson. Imaging* 27, 685–691. doi: 10.1002/jmri.21049

Jack, C. R. Jr., Bernstein, M. A., Borowski, B. J., Gunter, J. L., Fox, N. C., Thompson, P. M., et al. (2010). Update on the magnetic resonance imaging core of the Alzheimer's disease neuroimaging initiative. *Alzheimers Dement.* 6, 212–220. doi: 10.1016/j.jalz.2010.03.004

Jovicich, J., Czanner, S., Han, X., Salat, D., Van Der Kouwe, A., Quinn, B., et al. (2009). MRI-derived measurements of human subcortical, ventricular and intracranial brain volumes: reliability effects of scan sessions, acquisition sequences, data analyses, scanner upgrade, scanner vendors and field strengths. *Neuroimage* 46, 177–192. doi: 10.1016/j.neuroimage.2009.02.010

Karlik, S. J., Heatherley, T., Pavan, F., Stein, J., Lebron, F., Rutt, B., et al. (1988). Patient anesthesia and monitoring at a 1.5-T MRI installation. *Magn. Reson. Med.* 7, 210–221. doi: 10.1002/mrm.1910070209

Kasahara, M., Menon, D. K., Salmond, C. H., Outtrim, J. G., Tavares, J. V., Carpenter, T. A., et al. (2011). Traumatic brain injury alters the functional brain network mediating working memory. *Brain Inj.* 25, 1170–1187. doi: 10.3109/02699052.2011.608210

Keator, D. B., Helmer, K., Steffener, J., Turner, J. A., Van Erp, T. G., Gadde, S., et al. (2013). Towards structured sharing of raw and derived neuroimaging data across existing resources. *Neuroimage* 82, 647–661. doi: 10.1016/j.neuroimage.2013.05.094

Kempton, M. J., Underwood, T. S., Brunton, S., Stylios, F., Schmechtig, A., Ettinger, U., et al. (2011). A comprehensive testing protocol for MRI neuroanatomical segmentation techniques: evaluation of a novel lateral ventricle segmentation method. *Neuroimage* 58, 1051–1059. doi: 10.1016/j.neuroimage.2011.06.080

Kinnunen, K. M., Greenwood, R., Powell, J. H., Leech, R., Hawkins, P. C., Bonnelle, V., et al. (2011). White matter damage and cognitive impairment after traumatic brain injury. *Brain* 134, 449–463. doi: 10.1093/brain/awq347

Kraus, M. F., Susmaras, T., Caughlin, B. P., Walker, C. J., Sweeney, J. A., and Little, D. M. (2007). White matter integrity and cognition in chronic traumatic brain injury: a diffusion tensor imaging study. *Brain* 130, 2508–2519. doi: 10.1093/brain/awm216

Kuceyeski, A., Maruta, J., Niogi, S. N., Ghajar, J., and Raj, A. (2011). The generation and validation of white matter connectivity importance maps. *Neuroimage* 58, 109–121. doi: 10.1016/j.neuroimage.2011.05.087

Lehmann, M., Douiri, A., Kim, L. G., Modat, M., Chan, D., Ourselin, S., et al. (2010). Atrophy patterns in Alzheimer's disease and semantic dementia: a comparison of FreeSurfer and manual volumetric measurements. *Neuroimage* 49, 2264–2274. doi: 10.1016/j.neuroimage.2009.10.056

Lima, E. A., Irimia, A., and Wikswo, J. P. (2006). "The magnetic inverse problem," in *The SQUID Handbook*, Vol. 2, eds J. Clarke, and A. I. Braginski (Weinheim, Germany: Wiley-VCH), 139–268. doi: 10.1002/9783527609956.ch10

Ling, J., Merideth, F., Caprihan, A., Pena, A., Teshiba, T., and Mayer, A. R. (2012). Head injury or head motion? Assessment and quantification of motion artifacts in diffusion tensor imaging studies. *Hum. Brain Mapp.* 33, 50–62. doi: 10.1002/hbm.21192

MacDonald, C. L., Dikranian, K., Bayly, P., Holtzman, D., and Brody, D. (2007). Diffusion tensor imaging reliably detects experimental traumatic axonal injury and indicates approximate time of injury. *J. Neurosci.* 27, 11869–11876. doi: 10.1523/JNEUROSCI.3647-07.2007

Manley, G. T., and Maas, A. I. (2013). Traumatic brain injury: an international knowledge-based approach. *J. Am. Med. Assoc.* 310, 473–474. doi: 10.1001/jama.2013.169158

Margulies, D. S., Bottger, J., Watanabe, A., and Gorgolewski, K. J. (2013). Visualizing the human connectome. *Neuroimage* 80, 445–461. doi: 10.1016/j.neuroimage.2013.04.111

McDowell, S., Whyte, J., and D'esposito, M. (1997). Working memory impairments in traumatic brain injury: evidence from a dual-task paradigm. *Neuropsychologia* 35, 1341–1353. doi: 10.1016/S0028-3932(97)00082-1

Meythaler, J. M., Peduzzi, J. D., Eleftheriou, E., and Novack, T. A. (2001). Current concepts: diffuse axonal injury-associated traumatic brain injury. *Arch. Phys. Med. Rehabil.* 82, 1461–1471. doi: 10.1053/apmr.2001.25137

Moeller, S., Yacoub, E., Olman, C. A., Auerbach, E., Strupp, J., Harel, N., et al. (2010). Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI. *Magn. Reson. Med.* 63, 1144–1153. doi: 10.1002/mrm.22361

Mori, S., and van Zijl, P. C. (2002). Fiber tracking: principles and strategies–a technical review. *NMR Biomed.* 15, 468–480. doi: 10.1002/nbm.781

Niethammer, M., Hart, G. L., Pace, D. F., Vespa, P. M., Irimia, A., Van Horn, J. D., et al. (2011). Geometric metamorphosis. *Med Image Comput. Comput. Assist. Interv.* 14, 639–646.

Palacios, E. M., Sala-Llonch, R., Junque, C., Roig, T., Tormos, J. M., Bargallo, N., et al. (2013). Resting-state functional magnetic resonance imaging activity and connectivity and cognitive outcome in traumatic brain injury. *J. Am. Med. Assoc. Neurol.* 70, 845–851. doi: 10.1001/jamaneurol.2013.38

Pandit, A. S., Expert, P., Lambiotte, R., Bonnelle, V., Leech, R., Turkheimer, F. E., et al. (2013). Traumatic brain injury impairs small-world topology. *Neurology* 80, 1826–1833. doi: 10.1212/WNL.0b013e3182929f38

Powell, S., Magnotta, V. A., Johnson, H., Jammalamadaka, V. K., Pierson, R., and Andreasen, N. C. (2008). Registration and machine learning-based automated segmentation of subcortical and cerebellar brain structures. *Neuroimage* 39, 238–247. doi: 10.1016/j.neuroimage.2007.05.063

Prastawa, M., Bullitt, E., and Gerig, G. (2004). A brain tumor segmentation framework based on outlier detection. *Med. Image Anal.* 8, 275–283. doi: 10.1016/j.media.2004.06.007

Prastawa, M., Bullitt, E., Moon, N., Van Leemput, K., and Gerig, G. (2003). Automatic brain tumor segmentation by subject specific modification of atlas priors. *Acad. Radiol.* 10, 1341–1348. doi: 10.1016/S1076-6332(03)00506-3

Rubinov, M., and Sporns, O. (2010). Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* 52, 1059–1069. doi: 10.1016/j.neuroimage.2009.10.003

Scannell, J. W., Blakemore, C., and Young, M. P. (1995). Analysis of connectivity in the cat cerebral cortex. *J. Neurosci.* 15, 1463–1483.

Scannell, J. W., and Young, M. P. (1993). The connectional organization of neural systems in the cat cerebral cortex. *Curr. Biol.* 3, 191–200. doi: 10.1016/0960-9822(93)90331-H

Sharon, D., Hamalainen, M. S., Tootell, R., and Halgren, E. (2009). The advantage of combining MEG and EEG: comparison to fMRI in focally-stimulated visual cortex. *Neuroimage* 36, 1225–1235. doi: 10.1016/j.neuroimage.2007.03.066

Skudlarski, P., Jagannathan, K., Calhoun, V. D., Hampson, M., Skudlarska, B. A., and Pearlson, G. (2008). Measuring brain connectivity: diffusion tensor imaging validates resting state temporal correlations. *Neuroimage* 43, 554–561. doi: 10.1016/j.neuroimage.2008.07.063

Stanley, M. L., Moussa, M. N., Paolini, B. M., Lyday, R. G., Burdette, J. H., and Laurienti, P. J. (2013). Defining nodes in complex brain networks. *Front. Comput. Neurosci.* 7:169. doi: 10.3389/fncom.2013.00169

Storti, S. F., Formaggio, E., Franchini, E., Bongiovanni, L. G., Cerini, R., Fiaschi, A., et al. (2012). A multimodal imaging approach to the evaluation of post-traumatic epilepsy. *MAGMA* 25, 345–360. doi: 10.1007/s10334-012-0316-9

Strangman, G. E., O’neil-Pirozzi, T. M., Supelana, C., Goldstein, R., Katz, D. I., and Glenn, M. B. (2010). Regional brain morphometry predicts memory rehabilitation outcome after traumatic brain injury. *Front. Hum. Neurosci.* 4:182. doi: 10.3389/fnhum.2010.00182

Strogatz, S. H. (2001). Exploring complex networks. *Nature* 410, 268–276. doi: 10.1038/35065725

Thirion, B., Tucholka, A., and Poline, J. (2010). “Parcellation schemes and statistical tests to detect active regions on the cortical surface,” in *Proceedings of the Nineteenth International Conference on Computational Statistics,* Paris.

Thompson, P. M., Hayashi, K. M., De Zubicaray, G., Janke, A. L., Rose, S. E., Semple, J., et al. (2003). Dynamics of gray matter loss in Alzheimer’s disease. *J. Neurosci.* 23, 994–1005.

Tijssen, R. H., Jansen, J. F., and Backes, W. H. (2009). Assessing and minimizing the effects of noise and motion in clinical DTI at 3 T. *Hum. Brain Mapp.* 30, 2641–2655. doi: 10.1002/hbm.20695

Ugurbil, K. (2012). The road to functional imaging and ultrahigh fields. *Neuroimage* 62, 726–735. doi: 10.1016/j.neuroimage.2012.01.134

Van Dijk, K. R., Sabuncu, M. R., and Buckner, R. L. (2012). The influence of head motion on intrinsic functional connectivity MRI. *Neuroimage* 59, 431–438. doi: 10.1016/j.neuroimage.2011.07.044

Van Horn, J. D., Irimia, A., Torgerson, C. M., Chambers, M. C., Kikinis, R., and Toga, A. W. (2012). Mapping connectivity damage in the case of phineas gage. *PLoS ONE* 7:e37454. doi: 10.1371/journal.pone.0037454

Van Horn, J. D., and Toga, A. W. (2009). Is it time to re-prioritize neuroimaging databases and digital repositories? *Neuroimage* 47, 1720–1734. doi: 10.1016/j.neuroimage.2009.03.086

Van Horn, J. D., and Toga, A. W. (2013). Human neuroimaging as a “Big Data” science. *Brain Imaging Behav.* doi: 10.1007/s11682-013-9255-y [Epub ahead of print].

Van Leemput, K., Maes, F., Vandermeulen, D., Colchester, A., and Suetens, P. (2001). Automated segmentation of multiple sclerosis lesions by model outlier detection. *IEEE Trans. Med. Imaging* 20, 677–688. doi: 10.1109/42.938237

Vincent, J. L., and Moreno, R. (2010). Clinical review: scoring systems in the critically ill. *Crit. Care* 14, 207. doi: 10.1186/cc8204

Wang, B., Prastawa, M. W., Awate, S. P., Irimia, A., Chambers, M. C., Vespa, P. M., et al. (2012). A patient-specific segmentation framework for longitudinal MR images of traumatic brain injury. *Proc. SPIE* 8314, 7. doi: 10.1117/12.911043

Weiner, M. W., Veitch, D. P., Aisen, P. S., Beckett, L. A., Cairns, N. J., Green, R. C., et al. (2012). The Alzheimer’s disease neuroimaging initiative: a review of papers published since its inception. *Alzheimers Dement.* 8, S1–S68. doi: 10.1016/j.jalz.2011.09.172

Wu, Y., Warfield, S. K., Tan, I. L., Wells Iii, W. M., Meier, D. S., Van Schijndel, R. A., et al. (2006). Automated segmentation of multiple sclerosis lesion subtypes with multichannel MRI. *Neuroimage* 32, 1205–1215. doi: 10.1016/j.neuroimage.2006.04.211

Zalesky, A., Fornito, A., Egan, G. F., Pantelis, C., and Bullmore, E. T. (2012). The relationship between regional and inter-regional functional connectivity deficits in schizophrenia. *Hum. Brain Mapp.* 33, 2535–2549. doi: 10.1002/hbm.21379

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 30 August 2013; accepted: 11 February 2014; published online: 26 February 2014.

Citation: Goh SYM, Irimia A, Torgerson CM and Van Horn JD (2014) Neuroinformatics challenges to the structural, connectomic, functional, and electrophysiological multimodal imaging of human traumatic brain injury. Front. Neuroinform. ***8****:19. doi: 10.3389/fninf.2014.00019*

This article was submitted to the journal Frontiers in Neuroinformatics.

TECHNOLOGY REPORT ARTICLE
published: 17 December 2013
doi: 10.3389/fninf.2013.00031

High-throughput neuro-imaging informatics

***Michael I. Miller*[1,2,3]*, Andreia V. Faria*[4]*, Kenichi Oishi*[4] *and Susumu Mori*[4]**

[1] *Center for Imaging Science, Johns Hopkins Whiting School of Engineering, The Johns Hopkins University, Baltimore, MD, USA*
[2] *Institute for Computational Medicine, Johns Hopkins School of Medicine and Whiting School of Engineering, The Johns Hopkins University, Baltimore, MD, USA*
[3] *Department of Biomedical Engineering, The Johns Hopkins University, Baltimore, MD, USA*
[4] *The Russell H. Morgan Department of Radiology and Radiological Science, The Johns Hopkins University School of Medicine, Baltimore, MD, USA*

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Davide Chicco, Politecnico di Milano, Italy

****Correspondence:***
Michael I. Miller, Center for Imaging Science, The Johns Hopkins University, Clark 301, 3400 N Charles St., Baltimore, MD 21218, USA
e-mail: mim@cis.jhu.edu

This paper describes neuroinformatics technologies at 1 mm anatomical scale based on high-throughput 3D functional and structural imaging technologies of the human brain. The core is an abstract pipeline for converting functional and structural imagery into their high-dimensional neuroinformatic representation index containing O(1000–10,000) discriminating dimensions. The pipeline is based on advanced image analysis coupled to digital knowledge representations in the form of dense atlases of the human brain at gross anatomical scale. We demonstrate the integration of these high-dimensional representations with machine learning methods, which have become the mainstay of other fields of science including genomics as well as social networks. Such high-throughput facilities have the potential to alter the way medical images are stored and utilized in radiological workflows. The neuroinformatics pipeline is used to examine cross-sectional and personalized analyses of neuropsychiatric illnesses in clinical applications as well as longitudinal studies. We demonstrate the use of high-throughput machine learning methods for supporting (i) cross-sectional image analysis to evaluate the health status of individual subjects with respect to the population data, (ii) integration of image and personal medical record non-image information for diagnosis and prognosis.

Keywords: neuro-imaging, neuroinformatics, computational anatomy, functional imaging

INTRODUCTION

Imaging is one of the most powerful medical tools for monitoring human health. In the era of personalized medicine, periodic checkups via whole body imaging, combined with routine medical screening, genetic information, and comparison with population data is expected to be key information for monitoring health status, pathological condition, and therapeutic effect. High-throughput imaging technologies are becoming ubiquitous, driven by the deployment of whole body high resolution MR, CT, and PET imaging devices. While huge personal MR/CT based data records are routinely being collected for cross-sectional and longitudinal examination of the progression of diseases as manifest via tumor growth or atrophic neurodegeneration, currently while this information is stored in the medical PACS, usually only linguistic diagnostic encoding from the physician is stored in the searchable patient record. Such a lack of direct feature representation of the dense structural and functional phenotype precludes its use for systematic medical analysis such as population statistics or cross-modality correlation. Contrast this to what is emerging in high-throughput genomics.

There are several reasons. Clearly, utilizing the information from dense imagery from a longitudinal study, for example, presents daunting challenges. High-resolution whole body CT scans at 0.5 mm resolution for full body coverage would generate gigabytes of data. Visual inspection by a radiologist is overwhelming at the original resolution. Most often the images are down-sampled or low-resolution images are acquired to accommodate the storage and retrieval challenges. Constructing a parsimonious encoding of the discriminating information presents a fundamental challenge. In high-dimensional spaces such as that represented by the millions of measurements generated by 3D imagers, parsimonious representation of the measurable structural and functional phenotype is essential.

Exploiting the maximum potential of the imagers or the associated scans appears impractical without some form of encoding, or extreme data reduction. Reduction of high-dimensional imagery to symbolic knowledge representations encoded via the informative discriminating dimensions is one of the holy-grails of image analysis, a field which has advanced dramatically in the past several decades. From our own school of Grenander's metric pattern (Grenander, 1993) has emerged the field of computational anatomy (CA) for medical image analysis (Grenander and Miller, 1998, 2007; Toga and Thompson, 2001; Miller et al., 2002; Thompson and Toga, 2002; Ardekani et al., 2009; Ashburner, 2009; Pennec, 2009). The organizing principle in CA is that while there are variations in human structure and function, representation of the evolutionarily stable organization of processing in human beings are to a great extent organized around the structural manifestation of the genotype, throughout what we term the structural or anatomical phenotype. The evolutionary process has been masterful in its conservation of neural processing and its apparent organization around the macroscopic scales of human anatomy. We assume throughout that while functional layout is highly variable and ultimately associated with cellular architecture, it is manifest at the macroscopic scale of the topological organization of human anatomy and is preserved in large part cross-sectionally. Striking examples include the tonotopic organization of the auditory system for representing the axes of complex spectral representation, the somatosensory and motor homunculus in sensory and motor cortex, and the conformal like representation of visual space in the visual field. In each case the spatial axis encodes the functional axis representation.

The fact that functional topography is supported via dense topologic correspondence to the anatomical coordinates is the basis of our personalization of atlas based neuroinformatics. The personalization step is accomplished via the construction of a positioning system for neuroinformatics termed DiffeoMaps. This is an infinite dimensional positioning system which we term a Geodesic Positioning System (GPS) (Miller et al., 2013b) transferring information between atlas or world coordinate systems and individualized coordinate systems. We term it geodesic positioning since the metric is constructed based on the shortest (geodesic) flow of diffeomorphisms which connect the coordinates (Miller et al., 2013b). Such a transfer of the atlas representation to the coordinates of the individual allows for the organization of the high-throughput medical image record into a high-dimensional "feature vector" or an "index." Indexing via DiffeoMaps is the essential reduction or parsing of the individual into metadata representations upon which the machine learning phase of high-throughput neuroinformatics may be applied. Shown in **Figure 1** is our overall solution for high-throughput neuroinformatics, which includes atlases, diffeomorphic mapping for position (GPS), reduction to a high-dimensional feature vector or index encoding the anatomical and functional phenotypes, and machine learning via supervised clustering. This paper examines (i) cross-sectional image analysis to evaluate the health status of individual subjects with respect to the population data, (ii) integration of image and non-image information for diagnosis and prognosis.

RELATED WORKS

The high-throughput Neuro-Imaging Informatics introduced in this article is based on three core technologies; deformable multi-modal brain atlases, geodesic positioning of meta-data or semantic labels via diffeomorphic image transformation, and machine learning algorithms, as detailed in the Materials and Methods below. The deformable multi-modal brain atlases have been developed in both the coordinate systems provided by Montreal Neurological Institute (MNI) and the International Consortium of Brain Mapping (ICBM), which is a multicenter effort known for the MRI database and various brain atlases (http://loni.usc.edu/ICBM/). The atlases developed through this consortium have been implemented in leading software packages for the functional and anatomical brain analyses, such as Statistical Parametric Mapping (SPM, http://www.fil.ion.ucl.ac.uk/spm/), FSL (http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/), MRICron (http://www.mccauslandcenter.sc.edu/mricro/mricron/), as well as our own MRIstudio (https://www.mristudio.org/). The primary uses of the MNI or ICBM atlases are to be a reference space for the voxel-based image analysis, in which statistical analyses are performed on each voxel after all images are normalized to the atlas space. This voxel-based approach has been widely used since it enables researchers to statistically analyze the whole brain with very high spatial specificity, and to report their findings on a standardized coordinate system. This approach also allows users to apply various types of anatomical parcellation maps, such as the automatic anatomical labeling atlas (AAL) (Tzourio-Mazoyer et al., 2002) and the LONI Probabilistic Brain Atlas (LPBA) (Shattuck et al., 2008) for quantifying gray matter functions and anatomy. Our deformable multi-modal brain atlases are extensions of these attempting to group voxels based on anatomical or functional units, through which features of each brain are preserved at 1 mm scale. While most of the "atlas-based"

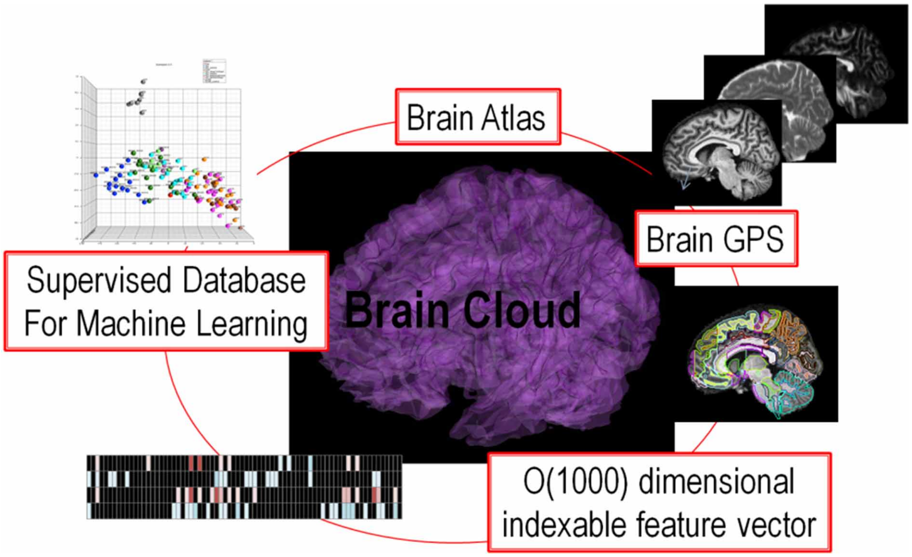


FIGURE 1 | Showing the core components of the high-throughput neuroinformatics pipeline including content (the atlas family), the personalization technology (GPS DiffeoMapping), and machine learning on the index or high-dimensional feature vector.

approaches previously have targeted the gray matter areas of single contrast images, the atlas used in our approach is multimodal, which means that the atlas consists of a set of images with different contrasts [e.g., T1- and T2-weighted images, Diffusion Weighted Imaging (DWI), Diffusion Tensor Imaging (DTI), and Susceptibility Weighted Image (SWI) contrasts] to allow multimodal image analysis of both gray and white matter structures in the common anatomical framework. The multimodal capability is supported by the Large Deformation Diffeomorphic Metric Mapping (LDDMM) methods that employ single and multichannel algorithms (Beg et al., 2005; Ceritoglu, 2008; Ceritoglu et al., 2009; Djamanakova et al., 2013), allowing for the incorporation of multiple imaging modalities while performing simultaneous mapping that maximally satisfies registration of the multiple modalities.

Another distinction we have made is to explicitly model both the geometric component of the atlas and associate to that the anatomical phenotype, simultaneously with the contrast component of the atlas which we generally associate to the function. This we do by providing a direct model in which the anatomical geometry carries the function, and demonstrate explicitly how to code via informatics both the anatomical geometry simultaneously with the functional contrasts. This forms the heart of our personalization via DiffeoMaps below. This allows us to directly generate classifiers and perform hypothesis generation about disease groups by both the anatomical phenotype as well as the contrast or function phenotype, and index them to different atlases. This can be contrasted to alternative approaches which use normalization viewing the geometric or anatomical phenotype as a nuisance parameter which is normalized out, like the affine group is removed rather than explicitly modeled.

Since the robustness of the machine learning framework to detect disease related anatomical and functional features of the brain has been demonstrated (Teipel et al., 2007; Hinrichs et al., 2011; Zhang et al., 2011), our approach is the generalizable extension toward high-throughput whole-brain multimodality analysis of heterogeneous brain conditions.

MATERIALS AND METHODS

ATLAS REPRESENTATION OF 1 mm STRUCTURAL—FUNCTIONAL CONTRASTS

The core of our high-throughput neuroinformatics technology is the conversion of the raw images into a structured, quantitative, and searchable high-dimensional feature vector. The basis for reduction to the numerical knowledge representation are the evolutionarily stable categorizations which neuroscientists have defined over the past decade. Our starting point is dense atlases of neuroanatomical structure and function indexed against age and group. We model the individual's imagery as an orbit under transformation of 1 mm scale coordinatized atlas information. **Figure 2** depicts our coordinatized human atlases demonstrating 3D anatomical information at different developmental stages (multi-dimensional) (Oishi et al., 2011c), different MR contrasts (multi-contrast) (Oishi et al., 2009), and varying coordinatized structural and functional definitions (Mori et al., 2013). The coordinate systems support MNI (Mazziotta et al., 1995, 2001) and Talairach (Talairach and Tournoux, 1988) coordinates as well as parcellations into different cortical areas as well as approximately 20 deep gray matter and 100 deep white matter structures all based on anatomical parcellation. The cortical partition includes structures such as parietal gyrus, frontal gyrus, pre-central gyrus, cuneus, lingual and others; the subcortical structures include amygdala, caudate, globus pallidus, hippocampus, putamen, thalamus, red nucleus, substantia nigra, hypothalamus, nucleus accumbens; the white matter structures include corticospinal, internal capsule, thalamic radiation, corona radiate, fornix, longitudinal fasciculus, corpus callosum, and others. Such a modern atlas also includes parcellations based on different anatomical and functional criteria such as cytoarchitecture, vascular territories, and anatomical and functional connectivity. This type of effort to parcellate the brain has been a subject of research based on histology (von Economo and Koskinas, 1925; Sarkisov et al., 1955; Mai et al., 1997; Schleicher et al., 1999; Tzourio-Mazoyer et al., 2002; Zilles et al., 2002) or MRI for the cortex (Lancaster et al., 2000; Mazziotta et al., 2001; Tzourio-Mazoyer et al., 2002; Hammers et al., 2003; Maldjian et al., 2003; Shattuck et al., 2008), white matter (Meyer et al., 1999; Mori et al., 2008; Oishi et al., 2008) and the whole brain (Fischl et al., 2002; Desikan et al., 2006; Oishi et al., 2009, 2011c, 2013).

PERSONALIZATION VIA DIFFEOMAPS AS A GEODESIC POSITIONING SYSTEM

Reduction to a high-dimensional feature vector which can be indexed requires us to model the high-throughput imagery. The underlying assumption of our model is that the meta-data representing the individual's structure and function is carried by the individual's coordinate systems, and there exists a structure preserving mapping which transforms the individual's coordinates into the stereotypical atlases. We term these transformations morphisms, these transformations form a group $\phi \in G$. The structure preserving morphisms provide correspondence between "charts" of the human brain as contained within atlas and the individual's coordinates. In this sense the morphisms provide a positioning system through their algebraic group action. Our group has come to call this the *metamorphism* model (Miller and Younes, 2001; Trouvé and Younes, 2005), organizing the structural and functional informatics, the images $I \in \Im$ into the transformation –image pair $[\phi(x), I(x)]$, $x \in X$ related via the algebraic pairing

$$\bullet : (\phi, I) \mapsto I' \doteq \phi \bullet I \in \Im. \quad (1)$$

In this model the morphisms denoted by $\phi(x)$, $x \in X$ carries the coordinatized contrast metadata imagery denoted by $I(x)$, $x \in X$.

Personalization occurs via smooth transformation of the atlas meta-data $\phi \cdot I_{\text{atlas}}$. For this we define a distance $\inf_{\phi} d(I, \phi \cdot I_{\text{atlas}})$ between the individual's representation and transformed atlas solving a variational problem for the coordinate (Dupuis et al., 1998; Beg et al., 2005; Ceritoglu et al., 2009) transformation. The correspondence between the individual and atlas is termed the "DiffeoMap," which provides an infinite dimensional positioning between atlas and world coordinates. This is in sharp contrast to the 7-dimensional similarity maps used in geographic positioning. To see this, the Eulerian velocities of Equation (2) below, while spatially smooth are a high-dimensional field, implying the Jacobian expressing first order transformation of coordinates in

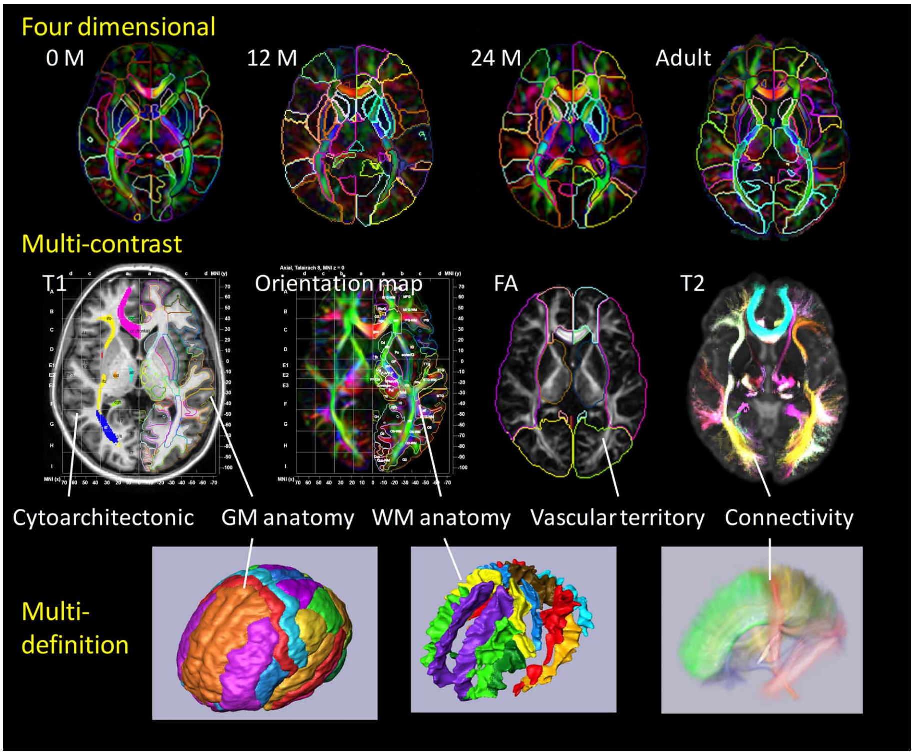


FIGURE 2 | Panels show a current brain atlas, including 3D anatomical information at different developmental stages (multi-dimension), different MR contrasts (multi-contrast), and different structural definitions. The coordinate systems include MNI and Talairach coordinates with the brain depicted as parcellated into multiple cortical and subcortical areas including deep gray and white matter structures based on anatomical features (anatomical parcellation) as well as functional parcellation based on cytoarchitecture, vascular territories, and anatomical and functional connectivity.

space allows the tissue to locally scale and twist while at the same time preserving relative organization.

Shown in **Figure 3** are instantiations of our structure-function metamorphosis model, including structural contrast imagery T1, orientation vector imagery such as DTI, metabolic contrast as measured via magnetic resonance spectroscopy and functional connectivity via resting-state fMRI (Faria et al., 2012).

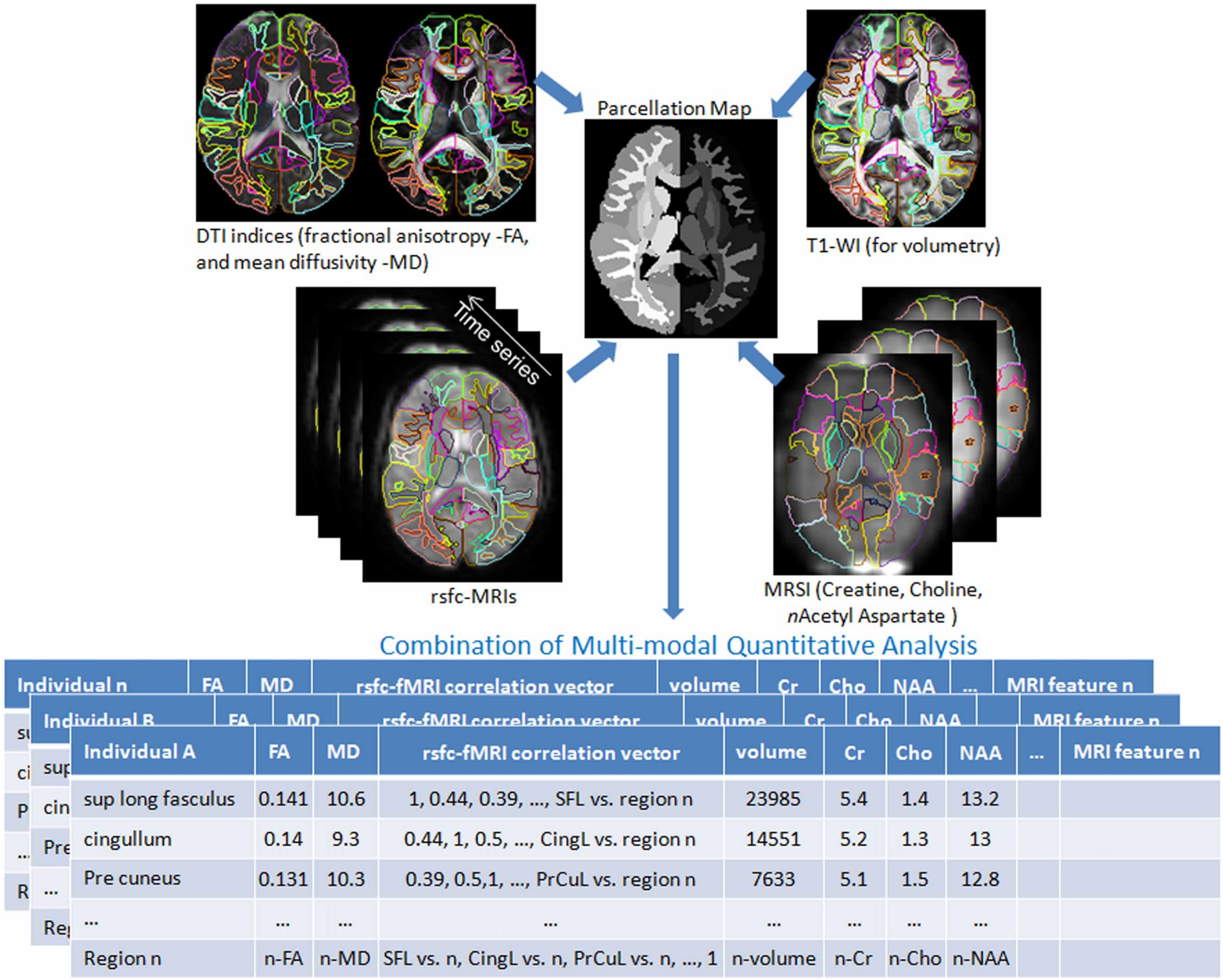
Each of the modalities has its own definition of the morphism acting on the meta-data of the contrast imagery explicating the algebra represented by •, specifically (i) for the submanifolds of subcortical structures, gyral curves and cortical surfaces the morphism acts $\phi \cdot x = \phi(x)$, (ii) for scalar imagery such as T1 the morphism acts via the inverse $\phi \cdot I = I \circ \phi^{-1}$, and (iii) for symmetric matrix-valued DTI (color in **Figure 1**) with eigen elements $\{\lambda_i, \phi_i\}$, the morphism acts to preserve the eigenvalues and determinant, rotating the eigenvectors $\phi \cdot I \doteq \left(\lambda_1 \hat{e}_1 \hat{e}_1^t + \lambda_2 \hat{e}_2 \hat{e}_2^t + \lambda_3 \hat{e}_3 \hat{e}_3^t\right) \circ \phi^{-1}$ with $\hat{e}_1 = \frac{(d\phi)\hat{e}_1}{\|(d\phi)\hat{e}_1\|}$, $\hat{e}_2 = \frac{(d\phi)\hat{e}_2 - \langle \hat{e}_1, (d\phi)\hat{e}_2 \rangle \hat{e}_1}{\sqrt{\|(d\phi)e_2\|^2 - \langle \hat{e}_1, (d\phi)\hat{e}_2 \rangle^2}}$, $\hat{e}_3 = \hat{e}_1 \times \hat{e}_2$, and $d\phi = \left(\frac{\partial \phi_i}{\partial x_j}\right)$ the 3 by 3 Jacobian matrix, with x denoting the vector cross-product.

THE HIGH-DIMENSIONAL FEATURE VECTOR AND MACHINE LEARNING

The scanners are the high-throughput devices generating the high-dimensional raw images of O(10,000,000) in complexity, and the pipeline converts it into a quantitative searchable feature vector $\{f = X_1, X_2, X_3, \ldots\}$ representing the individual at O(1000–10,000) complexity. Diffeomorphic GPS (Miller et al., 2013b) provides the basis for data reduction, since the anatomical structure phenotype is encoded by the morphisms and the meta-data of structure-function are encoded by the contrast imagery represented in atlas coordinates. The metamorphism model organizes the structural and functional informatics into the pair $[\phi(x), I(x)], x \in X$.

The GPS correspondences are diffeomorphisms, one-to-one and smooth mappings between coordinate systems $\phi : X \leftrightarrow$Y, $\phi(x), x \in X$ providing correspondences $\phi : I \leftrightarrow I_{\text{atlas}}$ between the individuals in the population and the atlas. The correspondences are generated as solutions of the classical Lagrangian flow equations, $\dot{\phi}_t = v_t(\phi_t)$, $t \in [0, 1]$ the time derivative of the flow v_t is termed the Eulerian velocity (Christensen et al., 1996). Constructing the DiffeoMaps occurs via the geodesic connection of one coordinate system to the other (Miller et al., 2006), solving for the geodesic connection between individual I and atlas $\phi \cdot I_{\text{atlas}}$ according to

$$\inf_{v_t,\, t \in [0,1] : \dot{\phi} = v(\phi),\, \phi_0 \cdot I_{\text{atlas}} = I_{\text{atlas}}} \int_0^1 \|v_t\|_V dt \text{ subject to } I = \phi_1 \cdot I_{\text{atlas}}. \tag{2}$$


Parcellation Map
DTI indices (fractional anisotropy -FA, and mean diffusivity -MD)
T1-WI (for volumetry)
Time series
rsfc-MRIs
MRSI (Creatine, Choline, nAcetyl Aspartate)
Combination of Multi-modal Quantitative Analysis
Individual n
Individual B

Individual A	FA	MD	rsfc-fMRI correlation vector	volume	Cr	Cho	NAA	…	MRI feature n
sup long fasculus	0.141	10.6	1, 0.44, 0.39, …, SFL vs. region n	23985	5.4	1.4	13.2		
cingullum	0.14	9.3	0.44, 1, 0.5, …, CingL vs. region n	14551	5.2	1.3	13		
Pre cuneus	0.131	10.3	0.39, 0.5,1, …, PrCuL vs. region n	7633	5.1	1.5	12.8		
…	…	…	…	…	…	…	…		
Region n	n-FA	n-MD	SFL vs. n, CingL vs. n, PrCuL vs. n, …, 1	n-volume	n-Cr	n-Cho	n-NAA		

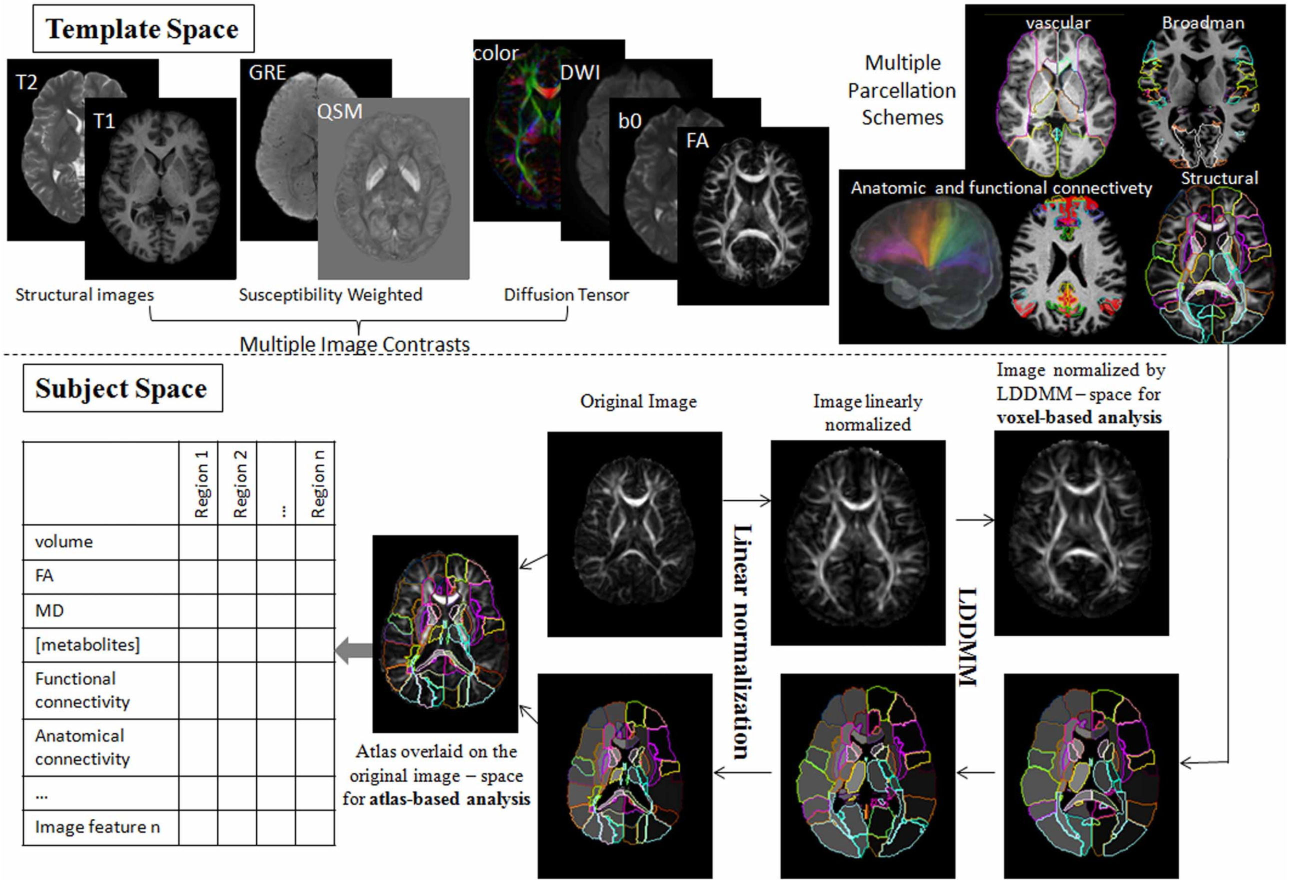
FIGURE 3 | Multiple image contrasts obtained from an individual using different MR pulse sequences. These multiple images are simultaneously parcellated into multiple structures, linking their coordinate systems (Parcellation Map). This procedure reduces the vast anatomical information into a parcellated series of approximately 200 structures and a series of MR contrast values that are the signature of each individual. DTI: Diffusion Tensor Image, T1-WI: T1 weighted images, rsfc: resting state functional connectivity, MRSI: Magnetic Resonance Spectroscopy Images. This pipeline is available at http://www.mricloud.org/ implementing multi-atlas parcellation (Tang et al., 2013b).

The geodesic connections are encoded via their initial tangent vector at the identity, denoted as $\nu = \nu_{t=0} \in V$. This forms the natural coordinate system of our GPS (Miller et al., 2013b). We have reduced the anatomical phenotype to a set of coordinates $\nu = \nu_{t=0} \in V$ centered at the atlas.

This is a natural representation of the anatomical or shape phenotype since the norm of the coordinates preserves the metric structure on the space of anatomies using this framework (Miller et al., 2006). The shortest flows connecting the template and individual coordinate systems define the metric in this space, the metric of Equation (2) is given by the integrated norm of the vector fields generating the morphisms. The reduction of the shape phenotype to these diffeomorphic connections we call *diffeomorphometry* (Miller et al., 2013b). At the 1 mm scale of MR imagery the anatomical phenotype is extremely sparse relative to the high-dimension of the initial data. For smooth imagery such as MRI linear functions of the vector fields, termed the shape momentum, are concentrated to the boundaries of the homogeneous subcomponents of the object (Miller et al., 2006; Qiu and Miller, 2008). At places in the image that are constant the shape is coded as zero. Plainly put, at the 1 mm scale gyral and subcortical regions of the MRI contrasts do not discriminate the cellular architecture.

Shown in **Figure 4** is an instantiation of our pipeline, depicting the personalization phase via DiffeoMap. The generation of the geodesic of Equation (2) for image matching via the solution of a quadratic variation problem on the vector field we call large deformation diffeomorphic metric mapping (LDDMM) (Beg et al., 2005). The modalities are shown in the top row in atlas coordinates with the DiffeoMap applied to the target showing the parcellation of target modalities shown in the bottom row.

Figure 5 shows a depiction of the subcortical neuronanatomy atlas as measured in 1 mm scale MR. The left panel shows


Template Space
T2
T1
GRE
QSM
color
DWI
b0
FA
Structural images
Susceptibility Weighted
Diffusion Tensor
Multiple Image Contrasts
Multiple Parcellation Schemes
vascular
Broadman
Anatomic and functional connectivety
Structural
Subject Space
Original Image
Image linearly normalized
Image normalized by LDDMM – space for voxel-based analysis
Linear normalization
LDDMM
Atlas overlaid on the original image – space for atlas-based analysis

	Region 1	Region 2	...	Region n
volume				
FA				
MD				
[metabolites]				
Functional connectivity				
Anatomical connectivity				
...				
Image feature n				

FIGURE 4 | Showing the pipeline starting with the modalities in atlas coordinates (top row) with the DiffeoMap applied to the target showing the parcellation of target modalities (bottom row). The algorithm used for solving for the multi-modality DiffeoMap is multi-modality LDDMM.

the atlas of 14 subcortical structures, amygdala (A, light blue), caudate (C), hippocampus (H, green), globus pallidus (PAL), putamen (PUT), ventricle (VL), thalamus (TH), each surface in the atlas containing order 1000 vertices. The set of structures correspond to an atlas generated from the population of healthy controls (HC) and Alzheimer's disease (AD) computed using the surface template estimation algorithm described in Ma et al. (2010). To demonstrate the sparcity of the anatomical phenotype at 1 mm scale, the geodesic correspondence between the atlas and a database of 250 subcortical brains were generated giving a coordinate identification of each element in the population, $I \sim v$, where v is the geodesic coordinate representation of the anatomy to the atlas. To understand the variation over the population, they were expanded via principle component analysis into a basis $v(f_1, f_2 \ldots) = \sum_i f_i U_i$; the f's are reduction of the anatomical phenotypes to the basis of eigenfunctions U. The sparsity of the anatomical phenotype was calculated across the population calculating the dimension required for encompassing 95% of the energy of subcortical variation. Generally each structure requires between 20 and 40 dimensions, with hippocampus and thalamus having the greatest shape variation within the population in terms of number of dimensions. The 95% variance cutoff as a function of dimensions for each structure is A20<PAL22<C25<PUT27,VL27<H30<THA40, the sparse subcortical shape phenotype at 1 mm scale is O(1000). The right panel shows the layout in the geodesic coordinate system of 250 of the anatomies (blue dots) in the first two geodesic dimensions with 20 of the brains shown explicitly.

This huge data reduction is noteworthy as it is the direct generalization of the sparsity of rigid body momentum which itself encodes translation and angular momentum to single 3-vectors, even though the inertia is extended over the entire object. Taking the midbrain as roughly 1/3 of the total brain volume of 2–4 Million voxels implies a data reduction of three orders of magnitude to O(1000).

CORTICAL, SUBCORTICAL AND WHITE MATTER PARCELLATION FEATURE VECTOR

The global positioning solution provides registered coordinates for the encoding of the target coordinates system into a parcellation corresponding to the anatomically defined partition of atlas coordinates in the 200 white and gray matter parcels. Denoting the atlas partition p_i, and since there can be as many as 7 MR contrast

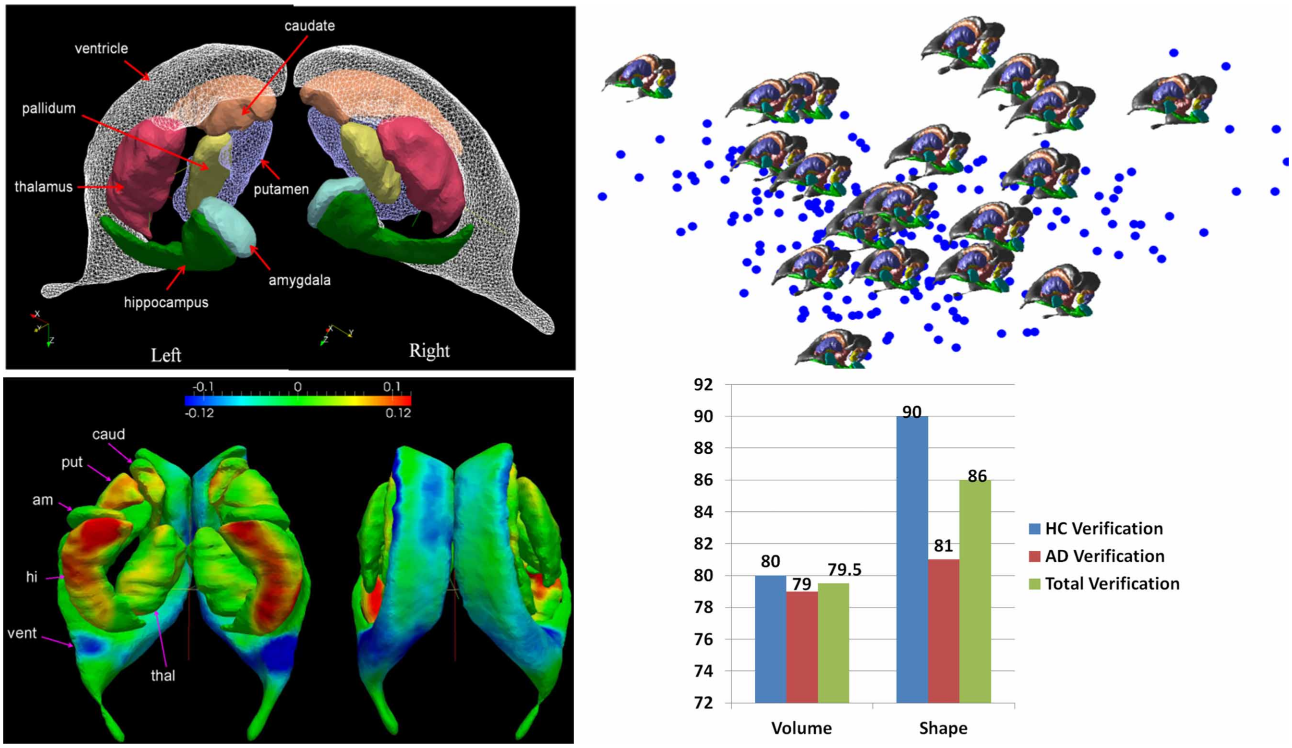


FIGURE 5 | Top row: Panel shows the left-right subcortical structures for human at 1 mm including amygdala (A, light blue), caudate (C), hippocampus (H, green), globus pallidus (PAL), putamen (PUT), ventricle (VL), thalamus (TH). The right panels shows the layout in the geodesic coordinate system of 250 of the anatomies (blue dots) in the first two dimensions with 20 of the brains shown explicitly with the basis dimensions on the order of 20–40 dimensions for each subcortical structure occupying 95% of the variatiance of anatomical variation with ordering A20<PAL22<C25<PUT27,VL27<H30<THA40. **Bottom row**: Shows the geodesic coordinates of the population (top right) relative to the atlas (top left) shown as a shape statistic computed by averaging over all geodesic mappings and computing the Jacobian of the tangent vector at the identity representing the anatomy. Bottom left shows the difference in the means $\mu^{HC} - \mu^{AD}$ superimposed on the template. The log-determinant of the Jacobian is shown, with red corresponding to shrinking and blue expansion. The bottom left panel depicts the hippocampus and amygdala are significantly red means large shrinkage relative to the contols, with the blue signaling the expansion of the ventricles. Bottom right panel shows a classifier based on three structures using only volume (left hand) and all the 20–40 dimensions of the anatomical phenotype encoded by the geodesic coordinates for hippocampus, amygdala, and ventricle. The images used for this analysis are a portion of a dataset published with the methodological detail (Tang et al., 2013a).

values including T1, T2, B0, trace, FA, spectroscopy, gives O(1000) features

$$f^c_{P_i} = \int_{P_i} I^c(x)dx, i = 1, \dots, 200, c = 1, .., 7. \qquad (3)$$

Shown in **Figure 6** are examples of the neuroinformatics parcellation which is transported via the personalization phase. **Figure 6A** shows the DiffeoMap personalization of the atlas into the coordinates of a spastic cerebral palsy patient with visually appreciable anatomical abnormalities (the color highlights the volume change larger than two standard deviations). The three rows show measurement results for volume, FA, and MD. Each column is an entry for one of the 200 anatomical structures. The top row represents anatomical information of each parcellated structure. In feature space, the neuroinformatics atlas supports both empirical means as well as empirical variances. Only features which demonstrate as outliers are depicted.

The bottom part, **Figure 6B**, shows an example of population data, in which the atlas partition of the anatomical phenotype for the listed structures (the bottom row in **Figure 6A**) are presented for 10 cerebral palsy patients (P3 is the individual shown in **Figure 6A**). All patients shared the same spastic phenotype with varying degree of motor impairment indicated by GMFCS scores. Abnormal parcellation volume values are presented by *z*-scores. At a glance, even though the patients were selected by similar clinical manifestations, a marked degree of anatomical variability can be recognized implying the importance of clustering on the spectrum of anatomical phenotype.

FUNCTIONAL MRI AND CONNECTIVITY MAPS IN ATLAS COORDINATES

Functional magnetic resonance imagery (fMRI) also provides ideal measurements for studying pairs of interactions in the brains. fMRI connectivity is based on empirical correlations of temporal responses between pairs of elements in the representation. **Figure 7** shows an example of empirical correlation of fMRI at lag-0 using the common atlas coordinate system to parcelate

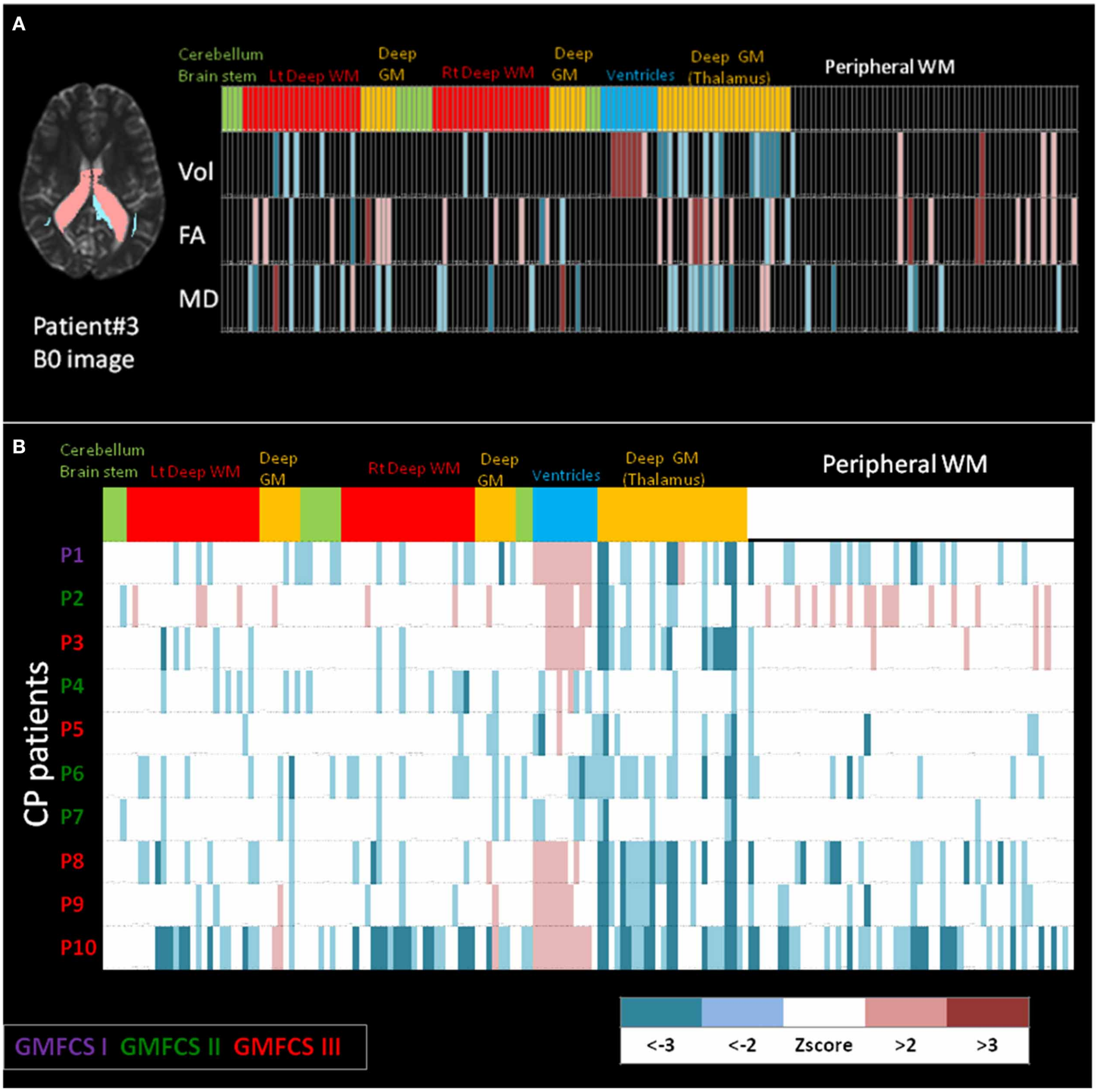


FIGURE 6 | (A) Feature vector from the personalization DiffeoMap correspondence between the atlas and an individual's coordinate system associated with focal disease category. Informatics partition with 200 structures including, volume, FA and MD values into peripheral white and gray matter, and deep white and gray matter structures. The features are color coded according to the statistics to depict color-coded outliers: WM: white matter, GM: gray matter. **(B)** Example of population data including 10 cerebral palsy patients with different prognoses in their motor disability. Informatics partition with 200 structures of volume for each patient is shown as 10 rows. The features are color coded according to *z* scores calculated based on normal control population. CP: cerebral palsy, GMFCS: gross motor function classification system.

symmetrically associated motor cortex areas plotting the resting-state MRI functions (rs-fMRI) (Tzourio-Mazoyer et al., 2002; Eickhoff et al., 2005; Achard et al., 2006; Hagmann et al., 2008; He et al., 2009; Wang et al., 2009).

Shown is the time series of the fMRI image modality $I^{\text{fMRI}}(x, t)$ integrated over the right and left motor cortex parcels. Notice the strong correlation depicted via the superposition of the red-blue time sequences. These highly correlated patches of tissue has resulted in the widely used ICA model in which the measured functional signal is the superposition of "networks,"

$$I^{\text{fMRI}}(x, t) = \sum_i f_i^{(t)} U_i^{(x)} \tag{4}$$

the U's playing the role of the resting-state networks. Working in the registered coordinates of the atlas allows for the construction of these resting state networks in the parcellations of the atlas by

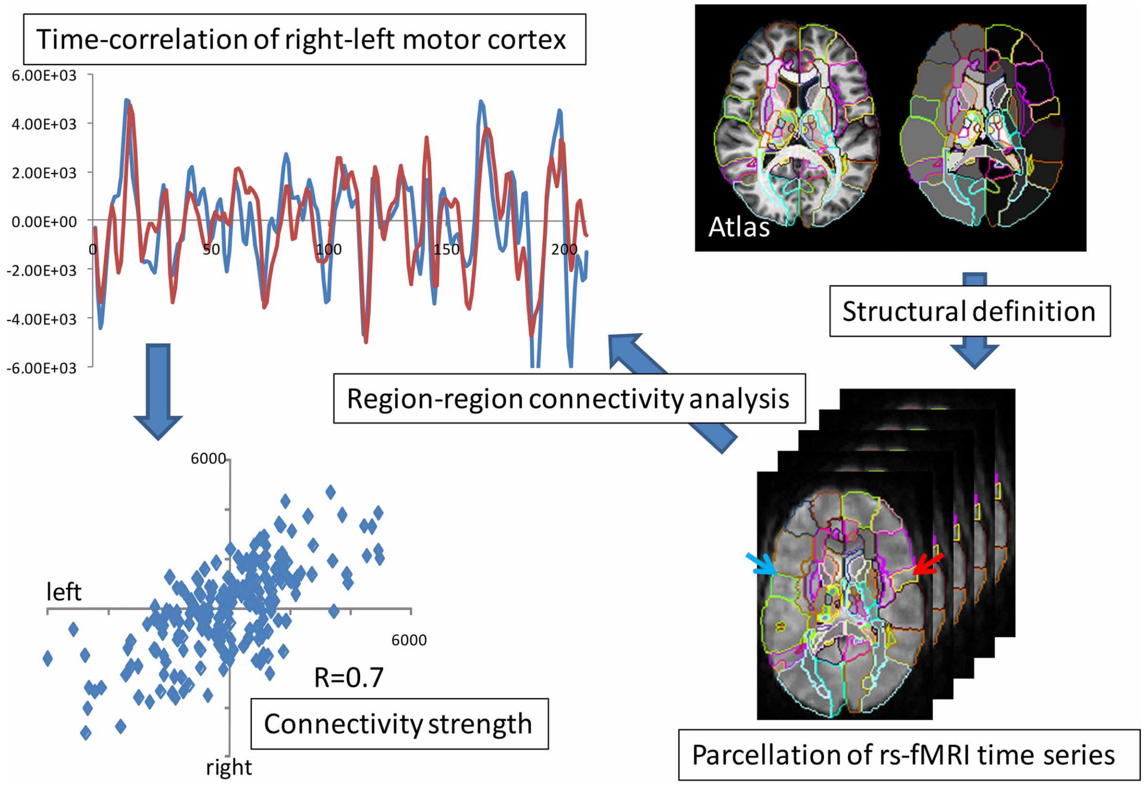


FIGURE 7 | fMRI parcellation based on resting state correlations. The **top panel** shows the overlap of the resting state functional signals integrated over the right-left motor parcellation; the **bottom panel** shows the value of blue and red time series over the 210 time points.

simply replacing the functional MR signal by it's parcellated representation $I_{P_i}(t) = \int_{P_i} I_{\mathrm{fMRI}}(x,\ t)dx, P_i = 1, \ldots 200$. The f's are the dimensions of the functional MRI signal representing 10–20 resting state dimensions added to the feature vector.

MACHINE LEARNING INVESTIGATION OF DISEASE-SPECIFIC PHENOTYPES

High spatial resolution is one of the most significant advantages of clinical MRI and its usefulness in studying pathological condition and detecting abnormalities. It seems clear from many studies that because of the noise versus signal tradeoff most detectable pathologies from MRI are signaled via small groups of spatially correlated voxel contrasts. Dimensionality reduction becomes the central methodology for MRI analysis in clinical applications. Combining unsupervised principal component analysis (PCA) along with supervised training, on the supervised group means under the common covariance model gives linear discriminant analysis (LDA).

Given m-length feature vectors, a collection n of them $\{f_j\}$, then PCA calculates the singular value decomposition (SVD) of the $m \times n$ matrix $F = (f_1, f_2, \ldots, f_n) = U\Sigma V^t$, where U is an $m \times m$ orthonormal matrix of vectors with $\sum$ diagonal with entries the singular values. The connection to least-squares and covariance modeling of Gaussian processes is that the left singular vectors $U = (U_1, \ldots, U_m)$ are the eigenfunctions of the empirical covariance FF^t; the set of diagonal entries squared of $\sum$ are the variances in the rotated independent representation of the left singular vectors. LDA then is the supervised version. Given groups of labeled feature vectors $\left\{f_j^g\right\}, g = 1, \ldots$ then each labeled group has a mean and covariance:

$$\mu^g = \sum\nolimits_{j=1}^{n_g} f_j^g/n_g, K^g = \sum\nolimits_{j=1}^{n_g} (f_j^g - \mu^g)(f_j^g - \mu^g)^t/n_g. \quad (5)$$

Then LDA is PCA on the group means μ^g using the common covariance $K = \sum_g K^g$. Quadratic discriminant analysis is a particular non-linear discriminant analysis (QDA) relaxing the common across groups covariance assumption. The high-dimensional structural and functional phenotypes are encoded via high-dimensional feature vectors. The classifiers are constructed from the cohorts of neuropsychiatric illnesses collected via the supervised training component. The crucial advantage of this approach is that the anatomical and structural phenotypes are indexed to the coordinates of the template. For the subcortical structures, for example, the anatomical phenotype is immediately reduced from a feature vector of dimension O(10,000,000), to the dimension of the surfaces which is O(10,000). Similarly, the functional feature is indexed over the anatomical substructures. This of course requires the notion of a template coordinate system which is centered in the population. Unlike other methods since we have explicitly modeled the anatomical and functional phenotypes, we can perform classification on both rather than viewing coordinate system transformation as a nuisance variable.

RESULTS

THE ANATOMICAL PHENOTYPE: IMAGE RETRIEVAL AND CLUSTERING

With an estimated 100 million scans every year in radiology, a huge amount of imaging data are generated every day, with these data stored in clinical Picture Archiving and Communication Systems (PACS) and are rarely used to support medical decision-making in cross-sectional examination of patient populations. Similar to the role of genomic and proteomic information for personalized medicine, anatomical phenotypes are fundamentally important for medical decision-making, yet often not systematically utilized in daily medical practice. While text-based patient records for retrieval of disease cohorts is commonly used, to utilize the anatomical phenotype for medical decision-making for individual patients we need to be able to use the patient image as the search key with the diagnostic label being the retrieved information. Using the high-dimensional feature vector without any diagnostic supervised labeling allows us to group and retrieve based on the structural phenotype.

Figure 8 shows an example of retrieval based on the anatomical phenotype, essentially delivering previously supervised cases with clinical information already in the data base. There are two types of information delivered in this analysis. For this we represent the anatomical variance of the population as shown in **Figure 5** for the subcortical structures to represent the coordinates of the anatomical position of the patient with respect to the atlas coordinate system relative to the population. This can be highly illustrative. For example, **Figure 8** shows healthy individuals as controls (green dots in the PCA plot of the structural volumes) and patients with two variants of Primary Progressive Aphasia (PPA), a neurodegenerative disease characterized by predominant and progressive deterioration in language in the absence of major change in personality, behavior or cognition other than praxis for at least two years. The *z*-score map in patient #3 reveals atrophy at the temporal left side that could be dubious at visual inspection only. In the addition, this subject is closer to other PPA patients than to the controls in the PCA plot, evidencing that the anatomical phenotype identified agrees with the clinical label.

Defining cohorts of similar patients is commonly done based on a host of features, including clinical behavioral and structural and functional phenotypes as measured in the functional and structural imagery. **Figure 8** examines clusters of cohorts based solely on the anatomical phenotype feature vector. The first principal component (PC1) accounts for global cortico-subcortical atrophy and ventricle enlargement, and mainly segregates age-matched controls from the PPA population. The segregation between two PPA variants (Semantic-SvPPA and Logopenic-LvPPA) is driven by severe and global atrophy of deep areas in SvPPA (PC2) and the predominant fronto-parietal atrophy in LvPPA, with relative preservation of temporal areas, particularly left, when compared with SvPPA (PC3). This agrees with past anatomical qualitative description of these populations. The existence of "outliers" corresponding to patients or controls surrounded by subjects of different labels are due to the singular anatomical features of these subjects. This type of analysis provides a platform for hypothesis-free comprehensive characterization of anatomical phenotype. Such quantitative analysis allows the investigation of various anatomy-associating factors, such as disease progression and functional outcomes, in a systematic manner.

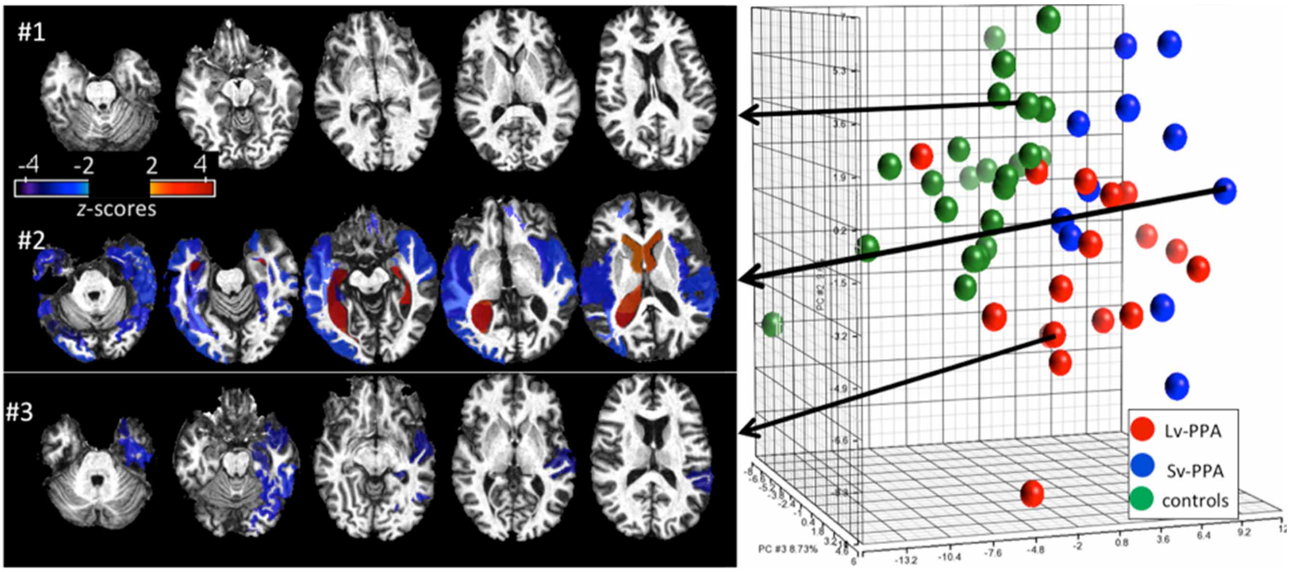


FIGURE 8 | Representation of degrees of regional atrophy as *z*-scores to support diagnosis (left panel). In cross-sectional studies on patients with similar diagnostic criteria, the patterns of atrophy from populations can be integrated with clinical information providing diagnostic and prognostic information. Clustering on the dimensions of the anatomical phenotype (**right panel**). The PCA plot contains the volumes of 200 brain structures in 24 healthy controls and 28 PPA patients. Shown are groupings according to the anatomical features associated to the clinical labels (controls, SvPPA, LvPPA).

DISEASE QUANTIFICATION OF ANATOMICAL PHENOTYPE VIA GEOODESIC COORDINATES

The GPS provides geodesic coordinates for representing every element in the population relative to the templates. We have examined machine learning on the subcortical structure coordinates shown in **Figure 5**. In the ADNI (Mueller et al., 2005) project there is extensive diagnostic supervised labeling enabling group based discriminations such as LDA/QDA for cross-sectional study of cohorts in dementia. A total of 385 subjects were segmented into their subcortical structures and lateral ventricle using FreeSurfer (Fischl et al., 2002) based on the analyses published by the Dale group (Fennema-Notestine et al., 2009). There were a total of 210 HC and 175 subjects with AD. To illustrate the average differences between the healthy control and Alzheimer's disease populations a HC-template surface and an AD-template surface was generated representing the center of each of the populations. We compared these two, HC-only and AD-only, template surfaces from the two different populations, and which are represented in geodesic coordinates relative to the overal template representing both HA-AD populations via the two class means μ^{HC}, μ^{AD}. To visualize these as shapes we calculated a scalar field corresponding to the log-determinant of the Jacobian of the map between the two averages, and visualized it on the template surface generated from the HC population generated using the algorithm described in Ma et al. (2008, 2010). This scalar field measures how much expansion/atrophy at each vertex of averaged surface from AD compared to that from HC in the logarithmic scale: i.e., positive value corresponds to surface expansion in the AD averaged surface at a particular location, while negative value denotes surface atrophying. The bottom left panel of **Figure 5** shows the mean differences between the two populations (bottom left), and is a visualization of one of the direction vectors in the Fischer discriminant the difference between the means $\mu^{HC} - \mu^{AD}$, shown as a plot of the Jacobian determinant. The color red represents the determinant being less then one, corresponding to shrinkage. The blue color corresponds to expansion. We see most of the shape change occurring at the ventricle expansion and the hippocampus and amygdala shrinkage.

The bottom right panel of **Figure 5** shows the result of building classifiers via machine learning whose discriminating dimensions are encoded in the picture in the lower left panel. We constructed the LDA and quadratic QDA classifiers using one of the *Leave-One-Out Cross-Validation* resampling method generating 385 LDA classifiers, testing on one of the subjects treating them as the testing data, and constructing the LDA class means μ^{HC}, μ^{AD} from the other 384 subjects (Tang et al., 2013a). For the two class problem the discriminating direction resulting from LDA on the geodesic coordinates is the projection of the differences in the means according to $K^{-1}(\mu^{HC} - \mu^{AD})$ on the common covariance. The Bayes classifier for the two class problem becomes a comparison to a threshold of the inner product of the feature vector on the discrimating direction:

$$f^t K^{-1} \left(\mu^{HC} - \mu^{AD}\right) \overset{HC}{\underset{AD}{\gtrless}} \theta. \tag{6}$$

As shown in **Figure 5** we find uniformly, as depicted by the blue bars in the classifier diagram, that the shape dimensions associated with the subcortical structures are significantly more discriminating then the volumes, generally reducing the errors in discrimination by more than 10%. The significant dimensions in volume and shape are associated with hippocampus and amygdala agreeing with previous results (Qiu et al., 2009). The specificity and sensitivity based on using the PCA shape dimensions in the feature vector for these three subcortical structure phenotypes is 90 and 81%, respectively. This is consistent with recent findings in another preclinical dementia study (Miller et al., 2013a) in which the shape of the temporal lobe subcortical structures is more discriminating then volume measures as well as in a Huntingdon's disease study tracking caudate, putamen, and globus pallidus (Younes et al., 2012).

PERSONALIZED ANALYSES: PREDICTING FUTURE CONVERSION BASED ON WHITE MATTER STRUCTURAL REPRESENTATIONS

While most research studies are based on cross-sectional population-based analyses, clinical diagnosis is always based on single individuals. This is performed by visual inspection in daily radiological diagnosis, in which images are most likely analyzed in a structure-by-structure basis, not in a voxel-by-voxel basis. Atlas-based neuroinformatic analyses in terms of their aggregate scale of the feature vector is compatible with many current diagnostic practices. Interestingly, histopathological studies indicate that white matter is an excellent target for both the early diagnosis of AD and for monitoring disease progression, motivating the use of DTI for studying patients with AD (Brun and Englund, 1986; Englund et al., 1988; Meier-Ruge et al., 1992; Gunawardena and Goldstein, 2001; Pigino et al., 2003; Sjobeck et al., 2005; Stokin et al., 2005; Chevalier-larsen and Holzbaur, 2006; Oishi et al., 2011b). There are already a large number of cross-sectional group comparison studies reporting significant differences in DTI derived measurements between the patients and controls, suggesting that white matter damage may exist in the presymptomatic phase of AD (Rose et al., 2000; Kantarci et al., 2001; Medina et al., 2006; Ringman et al., 2007; Stahl et al., 2007; Zhou et al., 2008; Damoiseaux et al., 2009; Salat et al., 2010; Sexton et al., 2011). One of the important questions after group analyses is whether these findings can be applicable to each individual to predict future conversion from memory impairment without other cognitive deficits (amnestic mild cognitive impairment) to dementia caused by AD. This is important because the amnestic mild cognitive impairment is a clinical category including multiple diseases or conditions with different pathological background, and not all of them develop AD (Albert et al., 2011).

Figure 9 shows results of personalizing the cross-sectional atlas statistics to several patients. A weighted feature vector, which could separate AD from cognitively normal population, was created from training datasets including groups of patients and cognitively normal age-matched individuals using dimensionality reduction applied to the atlas feature vector, and then DiffeoMapped to each individual to calculate the projection onto each patient. Shown in **Figure 9** are examples of the prediction of the conversion to Alzheimer's dementia in successive followup. Notice this projection doesn't predict conversion from amnestic mild cognitive impairment to the dementia with Lewy body, which is another type of neurodegenerative dementia. This type of analysis requires a large database with

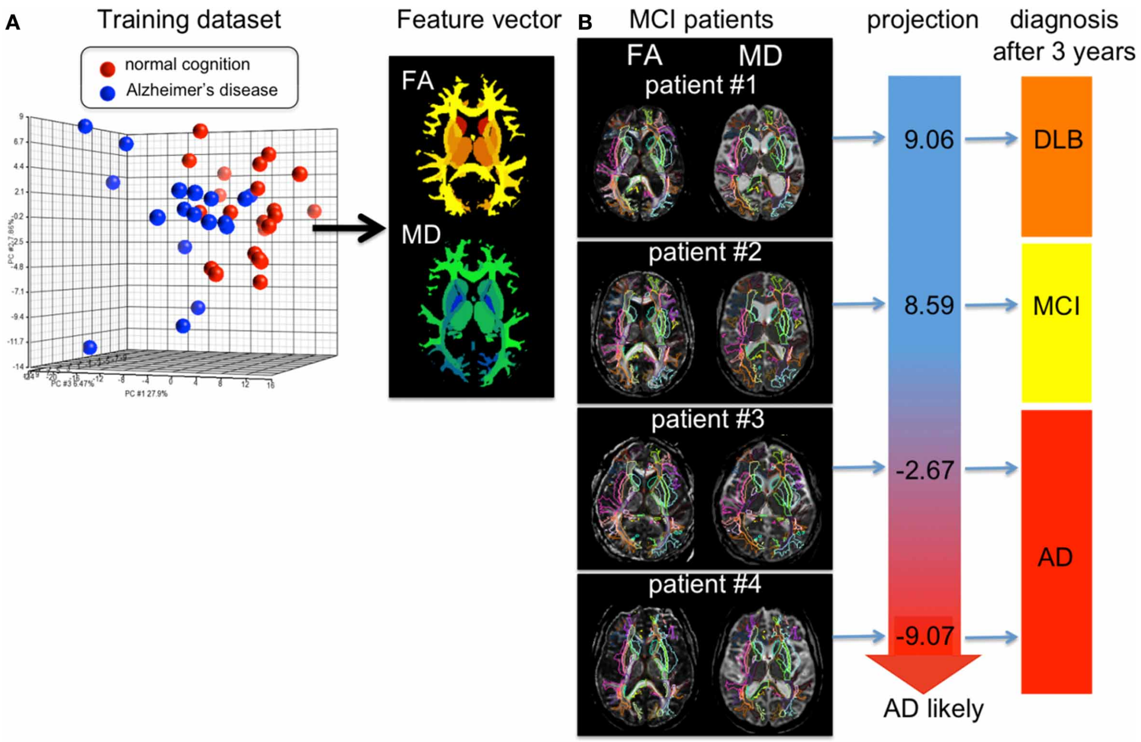


FIGURE 9 | (A) shows the result of PCA of the DTI derived measurements (FA = fractional anisotropy and MD = mean diffusivity) from 136 white matter areas and 12 deep gray matter structures. The first component was used as a diagnostic feature vector; the brighter area indicates more weighting to a degree of FA reduction and the cold brighter area indicate more weighting to a degree of MD increase to separate the AD group from the control group. **(B)** For individual images, the atlas was DiffeoMapped and projection to the feature vector was calculated. The projection well predicted early conversion from amnestic mild cognitive impairment (MCI) to Alzheimer's disease (AD), but did not predict conversion from MCI to the dementia with Lewy body (DLB). The DTIs used for this analysis are a portion of a dataset published in Oishi et al. (2011a).

longitudinal follow up which is an important current focus of our efforts.

DISCRIMINATING BETWEEN MULTIPLE DISEASES

The concept of group analysis in research studies assumes consistent locations of abnormalities, which does not hold for clinical situations, with heterogeneous patient populations and lack of an age-matched control group. The atlas-based neuroinformatics is compatible with the analysis of multiple diseases with different anatomical features. **Figure 10** shows the applicability of atlas-based neuroinformatics to capture anatomical features of multiple neurodegenerative diseases with known macroscopic anatomical alterations. To appropriately integrate diagnostic information to characterize the anatomical features related to each disease category, PCA and LDA were applied sequentially to a dataset consisting of 102 T1-weighted images from AD, primary progressive aphasia, Huntington's disease, hereditary spinocerebellar ataxia and normal control participants. These were parcellated based on the JHU-atlas [the images used for this analysis are a portion of a dataset published with the methodological detail (Qin et al., 2013)]. The weighted feature vectors efficiently captured known disease-specific anatomical alterations. For example, the medial temporal lobe and the parietal lobe were negatively weighted in the feature vector of AD to give a higher discriminant score for AD compared with other diseases and the control group. It should be noted that ventricular enlargement was not emphasized in the feature vector, although it was seen in most of the AD patients. Ventricular enlargement has been regarded as one of the disease-related features in past studies based on a cross-sectional comparison between AD and a control group, but seems to contain less information for separating AD from other neurodegenerative diseases.

FUNCTIONAL MRI PHENOTYPES IN ATLAS COORDINATES

Shown in **Figure 11** are results from functional magnetic resonance imaging done in registered atlas coordinates. Given the accompanying structural T1 images the functional responses can be examined in atlas coordinates with hypotheses formed at the scale of the partition of the atlas. Shown is a comparison between 7 patients with stroke at deep gray matter (cortex is preserved) and age-paired HC. The intensity plot shows the average of Fisher-transformed correlations between the rs- fMRI time courses of each pair of 42 cortical regions in controls (bottom) and individuals with stroke (top). In general, correlations between temporal and frontal areas are the main source of differences between the groups.

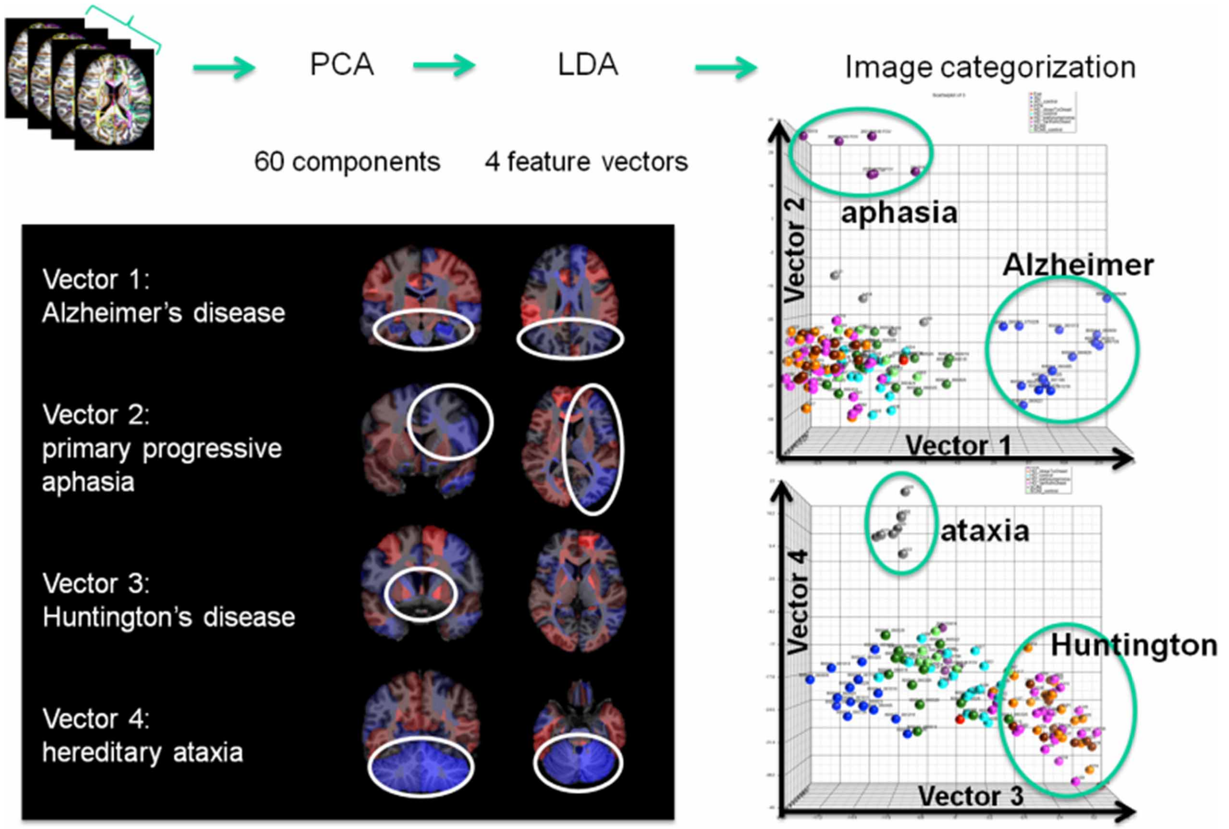


FIGURE 10 | Showing four clinically labeled disease categories of Alzheimer's Disease, aphasia, Huntington's, hereditary ataxia, and one control group upon which the anatomical features were learned including 60 PCA dimensions followed by supervised LDA delivering 4 loading vectors for discrimination. The clustering features in the high-dimensional index on the **right** are shown to correspond to anatomically meaningful shape representation shown in the **left panel**.

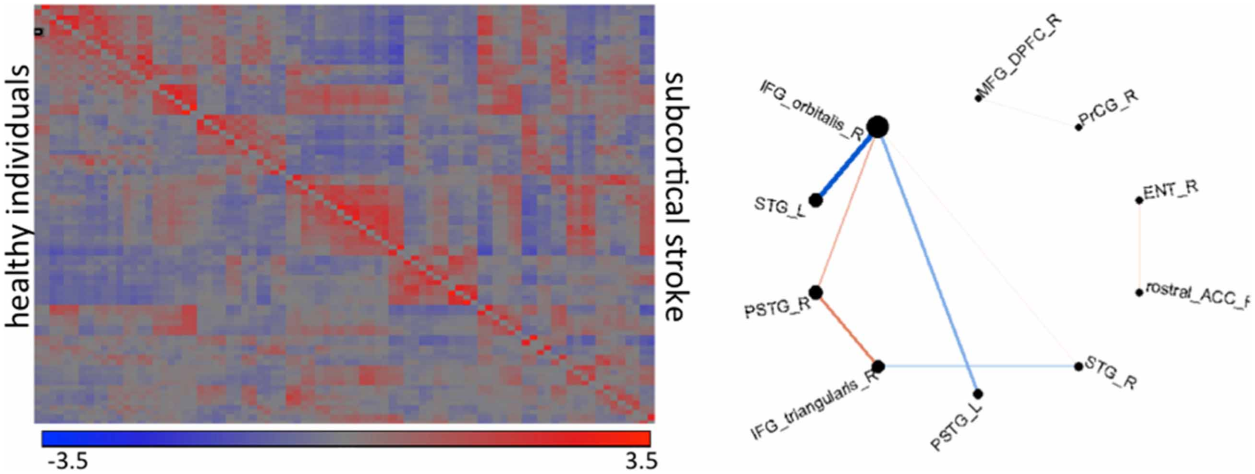


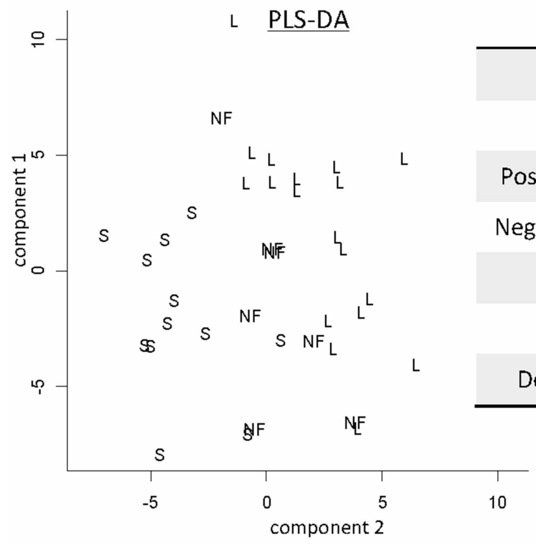
FIGURE 11 | The intensity plot shows average of Fisher-transformed 84 × 84 correlations of rs-fMRI response in atlas partition in individuals with subcortical stroke (superior to the diagonal) and controls (inferior to the diagonal). The diagram shows the connections that are different between groups ($p < 0.001$); the thickness of the lines is proportional to the ratio of the correlations (stroke/controls); blue are correlations with opposite signal between groups (positive—negative); red are those with same signal. R: right hemisphere, L: left hemisphere, IFG_orbitalis and IFG_triangularis: pars orbitalis and triangularis of the inferior frontal gyrus, MFG_DPFC: dorsal prefrontal pars of middle frontal cortex, PSTG and STG: posterior and medial pars of the superior temporal gyrus, rostral_ACC: rostral pars of the anterior cingulate gyrus, PrCG: pre-central gyrus, Ent: entorhinal area.

CLINICAL INFORMATICS AND BEHAVIOR PHENOTYPES AND FUNCTIONAL PHENOTYPES

Shown in **Figure 12** are results of demonstrating high-throughput informatics used to classify individuals into clinical phenotypes based on functional MRI coupled to clinical behaviors. The 3 clinical variants of PPA (logopenic—Lv, semantic—Sv, and non-fluent—NFv) may differ in terms of disease progression and response to therapeutics. In the early stages of the disease, when some therapeutics are being tested and will hopefully be effective, the clinical tests are not always able to classify all the patients. In addition, although anatomical differences among these variants are reported at group level, the individual classification based on qualitative evaluation is not usually possible. High-throughput imaging informatics can contribute for individual classification. **Figure 12** contains the volumetric data of 120 parcellated areas from 37 PPA patients that were scanned when, in their majority, the variant diagnosis wasn't completely clear, based on clinical information only. Our classification model, created using partial least squares—discriminant analysis (PLS-DA) and volumetric features (120 areas) demonstrated reasonable accuracy on predicting the variant diagnosis with a significant (higher than "by-chance") *p*-value, both when tested by bootstrapping or by external testing sample. The detection prevalence is low, particularly in the smallest group (NFv) with the sample size needed to be increased.

High-throughput informatics is also an effective tool to scrutinize anatomical-functional/ behavioral correlations. Much of the mapping of brain functions has been via lesion based studies, by relating regions affected by a stroke or trauma, for example, with the functional deficit. Lesion-based studies, however, have significant limitations such as (i) areas most strongly associated with the deficit depend on the vulnerability to ischemia/trauma (ii) determining the part of the lesion which is responsible for the deficit is difficult, or whether it represents a reorganization of cognitive networks that are less efficient, and (iii) the challenge of determining the proportion of changes leading to functional recovery, more than functional loss, (iv) the lack of multiple parameters, local or widespread, that might be concomitantly affected and whose interaction might correlate with the deficit.

Shown in the **Figure 13** is the application of quantitative analysis to assess anatomical-functional correlations in progressive disease models that affect specific functions (such as PPA, that affects primarily language) carried out by investigating the pattern of errors and their relationship to cortical impairment. It shows correlations between regional volumes and PPA patients' performance in a Naming test (Race et al., 2013). This type of anatomical-behavioral analysis provides a better understanding of the relationship between cognitive processes and regions necessary for particular aspects of processing. In more practical terms, we can use this information to monitor the disease progression,



Accuracy (Acc) : 0.9459
95% Confidence interval : (0.8181, 0.9934)
No Information Rate (NIR) : 0.4865
P-Value [Acc > NIR] : 2.065e-09
Kappa : 0.9125

	Lv	Sv	NFv
Sensitivity	1	0.9167	0.8571
Specificity	0.9474	1	0.9667
Positive Predictive Value	0.9474	1	0.8571
Negative Predictive Value	1	0.9615	0.9667
Prevalence	0.4373	0.3243	0.1892
Detection Rate	0.4865	0.2973	0.1622
Detection Prevalence	0.5135	0.2973	0.1892

Confusion Matrix

		Reference		
		L	S	NF
Prediction	L	18	0	1
	S	0	11	0
	NF	0	1	6

FIGURE 12 | Showing partial least squares—discriminant analysis (PLS-DA) for classifying 37 individuals into PPA variants based on volumetric data of 120 parcellated areas.

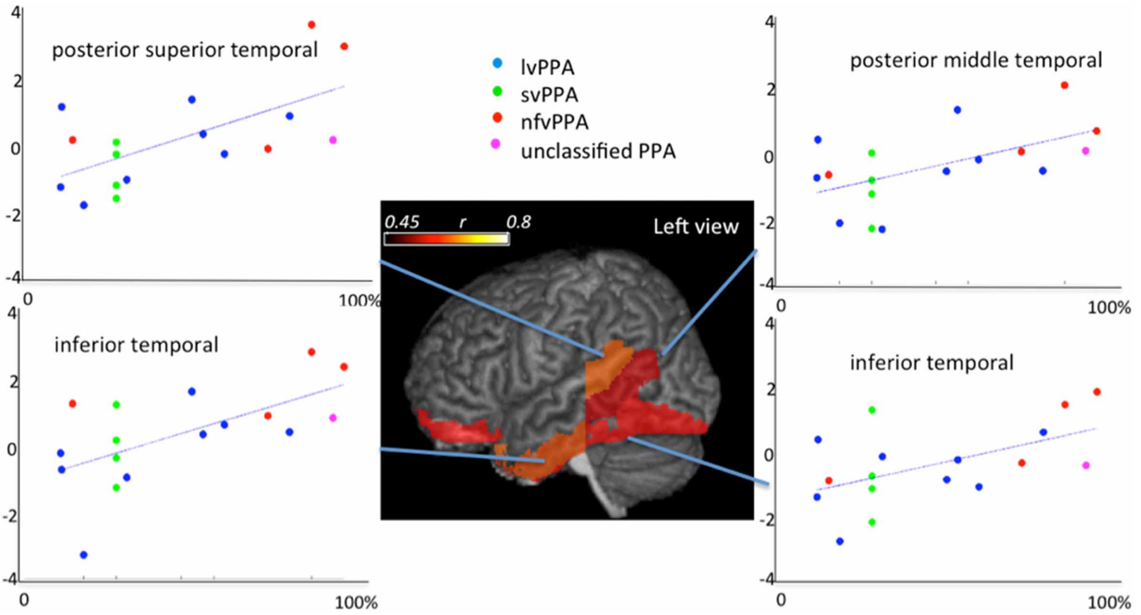


FIGURE 13 | Correlations between regional atlas anatomy (*z*-scores of volumes in *y* axis) and behavior (scores at Boston Naming Test—% of correctness in *x* axis) in individuals with Primary Progressive Aphasia—PPA. Regions with significant correlations are colored and the color scale represents the degree of correlation. These data includes part of the dataset used in Race et al. (2013).

or to categorize a clinical entity into more homogeneous groups, which can be meaningful if such subgroups express differences in prognosis or response to various treatments.

DISCUSSION

We have described neuroinformatics technologies at 1 mm anatomical scale based on high-throughput 3D functional and structural imaging technologies of the human brain. The core is the conversion of functional and structural imagery into their high-dimensional neuroinformatic representations index containing O(1000–10,000) discriminating dimensions. The pipeline is based on advanced image analysis coupled to digital knowledge representations in the form of dense atlases of the human brain at gross anatomical scale. We demonstrate the integration of these high-dimensional representations with machine learning methods.

The neuroinformatics pipeline is used to examine cross-sectional and personalized analyses of neuropsychiatric illnesses in clinical applications as well as longitudinal studies. We have demonstrated the use of high-throughput machine learning methods for supporting (i) cross-sectional image analysis to evaluate the health status of individual subjects with respect to the population data, (ii) integration of image and non-image information for diagnosis and prognosis.

ACKNOWLEDGMENTS

We gratefully acknowledge Shoko Yoshida for her efforts in the informatics, and Xioaying Tang for her efforts on the classifiers shown in **Figure 5** and Daniel Tward for his work on the geodesic coordinate and variability representation of subcortical structures. This work was supported by grants R01 MH056584, R24 HL085343, R01 EB000975, P50 MH071616, R01 EB001838, P41 EB015909, P41 EB015909, R01 EB008171, R01 MH084803, U01 AG033655, RR025053-01, U01 NS082085-01, R03 Eb014357, R21AG033774, R01HD065955, AHA12sdg12080169, and the Yousem Family Research Fund. This effort is being supported by the Computational Anatomy Science Gateway using the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

REFERENCES

Achard, S., Salvador, R., Whitcher, B., Suckling, J., and Bullmore, E. (2006). A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *J. Neurosci.* 26, 63–72. doi: 10.1523/JNEUROSCI.3874-05.2006

Albert, M. S., Dekosky, S. T., Dickson, D., Dubois, B., Feldman, H. H., Fox, N. C., et al. (2011). The diagnosis of mild cognitive impairment due to Alzheimer's disease: recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimers Dement.* 7, 270–279. doi: 10.1016/j.jalz.2011.03.008

Ardekani, S., Weiss, R. G., Lardo, A. C., George, R. T., Lima, J. A. C., Wu, K. C., et al. (2009). Computational method for identifying and quantifying shape features of human left ventricular remodeling. *Ann. Biomed. Eng.* 37, 1043–1054. doi: 10.1007/s10439-009-9677-2

Ashburner, J. (2009). Computational anatomy with the SPM software. *Magn. Reson. Imaging* 27, 1163–1174. doi: 10.1016/j.mri.2009.01.006

Beg, M. F., Miller, M. I., Trouve, A., and Younes, L. (2005). Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* 61, 139–157. doi: 10.1023/B:VISI.0000043755.93987.aa

Brun, A., and Englund, E. (1986). A white matter disorder in dementia of the Alzheimer type: a pathoanatomical study. *Ann. Neurol.* 19, 253–262. doi: 10.1002/ana.410190306

Ceritoglu, C. (2008). *Multichannel Large Deformation Diffeomorphic Metric Mapping and Registration of Diffusion Tensor Images*. Ph. D., The Johns Hopkins University.

Ceritoglu, C., Oishi, K., Li, X., Chou, M. C., Younes, L., Albert, M., et al. (2009). Multi-contrast large deformation diffeomorphic metric mapping for diffusion tensor imaging. *Neuroimage* 47, 618–627. doi: 10.1016/j.neuroimage.2009.04.057

Chevalier-larsen, E., and Holzbaur, E. L. (2006). Axonal transport and neurodegenerative disease. *Biochim. Biophys. Acta* 1762, 1094–1108. doi: 10.1016/j.bbadis.2006.04.002

Christensen, G. E., Rabbitt, R. D., and Miller, M. I. (1996). Deformable templates using large deformation kinematics. *IEEE Trans. Image Process.* 5, 1435–1447. doi: 10.1109/83.536892

Damoiseaux, J. S., Smith, S. M., Witter, M. P., Sanz-arigita, E. J., Barkhof, F., Scheltens, P., et al. (2009). White matter tract integrity in aging and Alzheimer's disease. *Hum. Brain Mapp.* 30, 1051–1059. doi: 10.1002/hbm.20563

Desikan, R. S., Segonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., et al. (2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage* 31, 968–980. doi: 10.1016/j.neuroimage.2006.01.021

Djamanakova, A., Faria, A. V., Hsu, J., Ceritoglu, C., Oishi, K., Miller, M. I., et al. (2013). Diffeomorphic brain mapping based on T1-weighted images: improvement of registration accuracy by multichannel mapping. *J. Magn. Reson. Imaging*, 37, 76–84. doi: 10.1002/jmri.23790

Dupuis, P., Grenander, U., and Miller, M. I. (1998). Variational problems on flows of diffeomorphisms for image matching. *Q. Appl. Math.* 56, 587–600.

Eickhoff, S. B., Stephan, K. E., Mohlberg, H., Grefkes, C., Fink, G. R., Amunts, K. et al. (2005). A new SPM toolbox for combining probabilistic cytoarchitectonic maps and functional imaging data. *Neuroimage* 25, 1325–1335. doi: 10.1016/j.neuroimage.2004.12.034

Englund, E., Brun, A., and Alling, C. (1988). White matter changes in dementia of Alzheimer's type. Biochemical and neuropathological correlates. *Brain* 111(pt 6), 1425–1439. doi: 10.1093/brain/111.6.1425

Faria, A. V., Joel, S. E., Zhang, Y., Oishi, K., Van zjil, P. C., Miller, M. I., et al. (2012). Atlas-based analysis of resting-state functional connectivity: evaluation for reproducibility and multi-modal anatomy-function correlation studies. *Neuroimage* 61, 613–621. doi: 10.1016/j.neuroimage.2012.03.078

Fennema-Notestine, C., McEvoy, L. K., Hagler, D. J. Jr., Jacobson, M. W., Dale, A. M., and The Alzheimer's Disease Neuroimaging, I. (2009). Structural neuroimaging in the detection and prognosis of pre-clinical and early AD. *Behav. Neurol.* 21, 3–12. doi: 10.3233/BEN-2009-0230

Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., et al. (2002). Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron* 33, 341–55. doi: 10.1016/S0896-6273(02)00569-X

Grenander, U. (1993). *General Pattern Theory: A Mathematical Study of Regular Structures.* Oxford, NY, Clarendon: Oxford University Press.

Grenander, U., and Miller, M. I. (1998). Computational anatomy: an emerging discipline. *Q. Appl. Math.* 56, 617–694.

Grenander, U., and Miller, M. I. (2007). *Pattern Theory: From Representation to Inference.* Oxford, NY: Oxford University Press.

Gunawardena, S., and Goldstein, L. S. (2001). Disruption of axonal transport and neuronal viability by amyloid precursor protein mutations in Drosophila. *Neuron* 32, 389–401. doi: 10.1016/S0896-6273(01)00496-2

Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Wedeen, V. J., et al. (2008). Mapping the structural core of human cerebral cortex. *PLoS Biol.* 6:e159. doi: 10.1371/journal.pbio.0060159

Hammers, A., Allom, R., Koepp, M. J., Free, S. L., Myers, R., Lemieux, L., et al. (2003). Three-dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe. *Hum. Brain Mapp.* 19, 224–47. doi: 10.1002/hbm.10123

He, Y., Wang, J., Wang, L., Chen, Z. J., Yan, C., Yang, H., et al. (2009). Uncovering intrinsic modular organization of spontaneous brain activity in humans. *PLoS ONE* 4:e5226. doi: 10.1371/journal.pone.0005226

Hinrichs, C., Singh, V., Xu, G., and Johnson, S. C. (2011). Predictive markers for AD in a multi-modality framework: an analysis of MCI progression in the ADNI population. *Neuroimage* 55, 574–589. doi: 10.1016/j.neuroimage.2010.10.081

Kantarci, K., Jack, C. R. Jr., Xu, Y. C., Campeau, N. G., O'brien, P. C., Smith, G. E., et al. (2001). Mild cognitive impairment and Alzheimer disease: regional diffusivity of water. *Radiology* 219, 101–107. doi: 10.1148/radiology.219.1.r01ap14101

Lancaster, J. L., Woldorff, M. G., Parsons, L. M., Liotti, M., Freitas, C. S., Rainey, L., et al. (2000). Automated Talairach atlas labels for functional brain mapping. *Hum. Brain Mapp.* 10, 120–131. doi: 10.1002/1097-0193(200007)10:3<120::AID-HBM30>3.0.CO;2-8

Ma, J., Miller, M. I., Trouve, A., and Younes, L. (2008). Bayesian template estimation in computational anatomy. *Neuroimage* 42, 252–261. doi: 10.1016/j.neuroimage.2008.03.056

Ma, J., Miller, M. I., and Younes, L. (2010). A bayesian generative model for surface template estimation. *Int. J. Biomed. Imaging* 2010. doi: 10.1155/2010/974957

Mai, J., Assheuer, J., and Paxinos, G. (1997). *Atlas of the Human Brain*. San Diego,CA: Academic.

Maldjian, J. A., Laurienti, P. J., Kraft, R. A., and Burdette, J. H. (2003). An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fMRI data sets. *Neuroimage* 19, 1233–1239. doi: 10.1016/S1053-8119(03)00169-1

Mazziotta, J., Toga, A., Evans, A., Fox, P., Lancaster, J., Zilles, K., et al. (2001). A probabilistic atlas and reference system for the human brain: international Consortium for Brain Mapping (ICBM). *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 356, 1293–1322. doi: 10.1098/rstb.2001.0915

Mazziotta, J. C., Toga, A. W., Evans, A., Fox, P., and Lancaster, J. (1995). A probabilistic atlas of the human brain: theory and rationale for its development. The International Consortium for Brain Mapping (ICBM). *Neuroimage* 2, 89–101. doi: 10.1006/nimg.1995.1012

Medina, D., Detoledo-Morrell, L., Urresta, F., Gabrieli, J. D., Moseley, M., Fleischman, D., et al. (2006). White matter changes in mild cognitive impairment and AD: a diffusion tensor imaging study. *Neurobiol. Aging* 27, 663–672. doi: 10.1016/j.neurobiolaging.2005.03.026

Meier-Ruge, W., Ulrich, J., Bruhlmann, M., and Meier, E. (1992). Age-related white matter atrophy in the human brain. *Ann. N.Y. Acad. Sci.* 673, 260–279. doi: 10.1111/j.1749-6632.1992.tb27462.x

Meyer, J. W., Makris, N., Bates, J. F., Caviness, V. S., and Kennedy, D. N. (1999). MRI-Based topographic parcellation of human cerebral white matter. *Neuroimage* 9, 1–17. doi: 10.1006/nimg.1998.0383

Miller, M. I., and Younes, L. (2001). Group actions, homeomorphisms, and matching: a general framework. *Int. J. Comput. Vis.* 41, 61–84. doi: 10.1023/A:1011161132514

Miller, M. I., Trouve, A., and Younes, L. (2002). On the metrics and Euler-Lagrange equations of computational anatomy. *Annu. Rev. Biomed. Eng.* 4, 375–405. doi: 10.1146/annurev.bioeng.4.092101.125733

Miller, M. I., Trouve, A., and Younes, L. (2006). Geodesic shooting for computational anatomy. *J. Math. Imaging Vis.* 24, 209–228. doi: 10.1007/s10851-005-3624-0

Miller, M. I., Younes, L., Ratnanather, J. T., Brown, T., Trinh, H., Postell, E., et al. (2013a). The diffeomorphometry of temporal lobe structures in preclinical Alzheimer's disease. *Neuroimage* 3, 352–360. doi: 10.1016/j.nicl.2013.09.001

Miller, M. I., Younes, L., and Trouvé, A. (2013b). Diffeomorphometry and geodesic positioning systems for human anatomy. *Technology* 2. Doi: 10.1142/S2339547814500010

Mori, S., Oishi, K., Faria, A. V., and Miller, M. I. (2013). Atlas-based neuroinformatics via mri: harnessing information from past clinical cases and quantitative image analysis for patient care. *Annu. Rev. Biomed. Eng.* 15, 71–92. doi: 10.1146/annurev-bioeng-071812-152335

Mori, S., Oishi, K., Jiang, H., Jiang, L., Li, X., Akhter, K., et al. (2008). Stereotaxic white matter atlas based on diffusion tensor imaging in an ICBM template. *Neuroimage* 40, 570–582. doi: 10.1016/j.neuroimage.2007.12.035

Mueller, S. G., Weiner, M. W., Thal, L. J., Petersen, R. C., Jack, C. R., Jagust, W., et al. (2005). Ways toward an early diagnosis in Alzheimer's disease: the Alzheimer's Disease Neuroimaging Initiative (ADNI). *Alzheimers Dement.* 1, 55–66. doi: 10.1016/j.jalz.2005.06.003

Oishi, K., Akhter, K., Mielke, M., Ceritoglu, C., Zhang, J., Jiang, H., et al. (2011a). Multi-modal MRI analysis with disease-specific spatial filtering: initial testing to predict mild cognitive impairment patients who convert to Alzheimer's disease. *Front. Neurol.* 2:54. doi: 10.3389/fneur.2011.00054

Oishi, K., Mielke, M. M., Albert, M., Lyketsos, C. G., and Mori, S. (2011b). DTI analyses and clinical applications in Alzheimer's disease. *J. Alzheimers Dis.* 26(Suppl. 3), 287–296. doi: 10.3233/JAD-2011-0007

Oishi, K., Mori, S., Donohue, P. K., Ernst, T., Anderson, L., Buchthal, S., et al. (2011c). Multi-contrast human neonatal brain atlas: application to normal neonate development analysis. *Neuroimage* 56, 8–20. doi: 10.1016/j.neuroimage.2011.01.051

Oishi, K., Faria, A., Jiang, H., Li, X., Akhter, K., Zhang, J., et al. (2009). Atlas-based whole brain white matter analysis using large deformation diffeomorphic metric mapping: application to normal elderly and Alzheimer's disease participants. *Neuroimage* 46, 486–499. doi: 10.1016/j.neuroimage.2009.01.002

Oishi, K., Faria, A. V., Yoshida, S., Chang, L., and Mori, S. (2013). Quantitative evaluation of brain development using anatomical MRI and diffusion tensor imaging. *Int. J. Dev. Neurosci.* 31, 512–524. doi: 10.1016/j.ijdevneu.2013.06.004

Oishi, K., Zilles, K., Amunts, K., Faria, A., Jiang, H., Li, X., et al. (2008). Human brain white matter atlas: identification and assignment of common anatomical structures in superficial white matter. *Neuroimage* 43, 447–457. doi: 10.1016/j.neuroimage.2008.07.009

Pennec, X. (2009). Statistical computing on manifolds: from Riemannian geometry to computational anatomy. *Emerg. Trends Visual Comput.* 5416, 347–386. doi: 10.1007/978-3-642-00826-9_16

Pigino, G., Morfini, G., Pelsman, A., Mattson, M. P., Brady, S. T., and Busciglio, J. (2003). Alzheimer's presenilin 1 mutations impair kinesin-based axonal transport. *J. Neurosci.* 23, 4499–4508.

Qin, Y. Y., Hsu, J. T., Yoshida, S., Faria, A. V., Oishi, K., Unschuld, P. G., et al. (2013). *Gross Feature Recognition of Anatomical Images based on Atlas Grid (GAIA): Using the Degree of Local Atlas-Image Segmentation Disagreement to Capture the Features of Anatomic Brain MRI*. NeuroiImage: Clinical, In Press.

Qiu, A., and Miller, M. I. (2008). Multi-structure network shape analysis via normal surface momentum maps. *Neuroimage* 42, 1430–1438. doi: 10.1016/j.neuroimage.2008.04.257

Qiu, A. Q., Fennema-Notestine, C., Dale, A. M., Miller, M. I., and Neuroimaging, A. S. D. (2009). Regional shape abnormalities in mild cognitive impairment and Alzheimer's disease. *Neuroimage* 45, 656–661. doi: 10.1016/j.neuroimage.2009.01.013

Race, D. S., Tsapkini, K., Crinion, J., Newhart, M., Davis, C., Gomez, Y., et al. (2013). An area essential for linking word meanings to word forms: evidence from primary progressive Aphasia. *Brain Lang.* 127, 167–176. doi: 10.1016/j.bandl.2013.09.004

Ringman, J. M., O'neill, J., Geschwind, D., Medina, L., Apostolova, L. G., Rodriguez, Y., et al. (2007). Diffusion tensor imaging in preclinical and presymptomatic carriers of familial Alzheimer's disease mutations. *Brain* 130, 1767–1776. doi: 10.1093/brain/awm102

Rose, S. E., Chen, F., Chalk, J. B., Zelaya, F. O., Strugnell, W. E., Benson, M., et al. (2000). Loss of connectivity in Alzheimer's disease: an evaluation of white matter tract integrity with colour coded MR diffusion tensor imaging. *J. Neurol. Neurosurg. Psychiatry* 69, 528–530. doi: 10.1136/jnnp.69.4.528

Salat, D. H., Tuch, D. S., Van der Kouwe, A. J., Greve, D. N., Pappu, V., Lee, S. Y., et al. (2010). White matter pathology isolates the hippocampal formation in Alzheimer's disease. *Neurobiol. Aging* 31, 244–56. doi: 10.1016/j.neurobiolaging.2008.03.013

Sarkisov, S. A., Filimonoff, I. N., Kononowa, E. P., Preobrachenskaja, I. S., and Kukuew, L. A. (1955). *Atlas of the Cytoarchitectonics of the Human Cerebral Cortex*. Moscow: Medgiz.

Schleicher, A., Amunts, K., Geyer, S., Morosan, P. and Zilles, K. (1999). Observer-independent method for microstructural parcellation of cerebral cortex: a quantitative approach to cytoarchitectonics. *Neuroimage* 9, 165–177. doi: 10.1006/nimg.1998.0385

Sexton, C. E., Kalu, U. G., Filippini, N., Mackay, C. E., and Ebmeier, K. P. (2011). A meta-analysis of diffusion tensor imaging in mild cognitive impairment and Alzheimer's disease. *Neurobiol. Aging*. 32, 2322.e5–2322.e18. doi: 10.1016/j.neurobiolaging.2010.05.019

Shattuck, D. W., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K. L., et al. (2008). Construction of a 3D probabilistic atlas of human cortical structures. *Neuroimage* 39, 1064–1080. doi: 10.1016/j.neuroimage.2007.09.031

Sjobeck, M., Haglund, M., and Englund, E. (2005). Decreasing myelin density reflected increasing white matter pathology in Alzheimer's disease–a neuropathological study. *Int. J. Geriatr. Psychiatry* 20, 919–926. doi: 10.1002/gps.1384

Stahl, R., Dietrich, O., Teipel, S. J., Hampel, H., Reiser, M. F., and Schoenberg, S. O. (2007). White matter damage in Alzheimer disease and mild cognitive impairment: assessment with diffusion-tensor MR imaging and parallel imaging techniques. *Radiology* 243, 483–492. doi: 10.1148/radiol.2432051714

Stokin, G. B., Lillo, C., Falzone, T. L., Brusch, R. G., Rockenstein, E., Mount, S. L., et al. (2005). Axonopathy and transport deficits early in the pathogenesis of Alzheimer's disease. *Science* 307, 1282–1288. doi: 10.1126/science.1105681

Talairach, J., and Tournoux, P. (1988). *Co-Planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System: an Approach to Cerebral Imaging*. New York, NY: Thieme.

Tang, X., Holland, D., Dale, A. M., Younes, L., Miller, M. I., and ADNI (2013a). Shape abnormalities of subcortical and ventricular structures in mild cognitive impairment and Alzheimer's disease: detecting, quantifying, and predicting. *Hum. Brain Mapp.*

Tang, X., Oishi, K., Faria, A. V., Hillis, A. E., Albert, M. S., Mori, S., et al. (2013b). Bayesian Parameter Estimation and Segmentation in the Multi-Atlas Random Orbit Model. *PLoS ONE* 8:e65591. doi: 10.1371/journal.pone.0065591

Teipel, S. J., Stahl, R., Dietrich, O., Schoenberg, S. O., Perneczky, R., Bokde, A. L., et al. (2007). Multivariate network analysis of fiber tract integrity in Alzheimer's disease. *Neuroimage* 34, 985–995. doi: 10.1016/j.neuroimage.2006.07.047

Thompson, P. M., and Toga, A. W. (2002). A framework for computational anatomy. *Comput. Visual Sci.* 5, 13–34. doi: 10.1007/s00791-002-0084-6

Toga, A. W., and Thompson, P. M. (2001). Maps of the brain. *Anat. Rec.* 265, 37–53. doi: 10.1002/ar.1057

Trouvé, A., and Younes, L. (2005). Metamorphoses through lie Group Action. *Foundations of Computational Mathematics* 5, 173–198. doi: 10.1007/s10208-004-0128-z

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., et al. (2002). Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage* 15, 273–289. doi: 10.1006/nimg.2001.0978

von Economo, C., and Koskinas, G. N. (1925). *Die Cytoarchitektonik der Hirnrinde des Erwachsenen Menschen*. Berlin: Springer.

Wang, J., Wang, L., Zang, Y., Yang, H., Tang, H., Gong, Q., et al. (2009). Parcellation-dependent small-world brain functional networks: a resting-state fMRI study. *Hum. Brain Mapp.* 30, 1511–1523. doi: 10.1002/hbm.20623

Younes, L., Ratnanather, J. T., Brown, T., Aylward, E., Nopoulos, P., Johnson, H., et al. (2012). Regionally selective atrophy of subcortical structures in prodromal HD as revealed by statistical shape analysis. *Hum. Brain Mapp*. doi: 10.1002/hbm.22214

Zhang, D., Wang, Y., Zhou, L., Yuan, H., and Shen, D. (2011). Multimodal classification of Alzheimer's disease and mild cognitive impairment. *Neuroimage* 55, 856–867. doi: 10.1016/j.neuroimage.2011.01.008

Zhou, Y., Dougherty, J. H. Jr., Hubner, K. F., Bai, B., Cannon, R. L., and Hutson, R. K. (2008). Abnormal connectivity in the posterior cingulate and hippocampus in early Alzheimer's disease and mild cognitive impairment. *Alzheimers Dement.* 4, 265–270. doi: 10.1016/j.jalz.2008.04.006

Zilles, K., Palomero-Gallagher, N., Grefkes, C., Scheperjans, F., Boy, C., Amunts, K., et al. (2002). Architectonics of the human cerebral cortex and transmitter receptor fingerprints: reconciling functional neuroanatomy and neurochemistry. *Eur. Neuropsychopharmacol.* 12, 587–599. doi: 10.1016/S0924-977X(02)00108-6

Conflict of Interest Statement: Susumu Mori and Michael I. Miller own Anatomy Works, with Susumu Mori serving as its CEO. This arrangement is being managed by Johns Hopkins University in accordance with its conflict of interest policies. The other authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 16 September 2013; accepted: 04 November 2013; published online: 17 December 2013.

Citation: Miller MI, Faria AV, Oishi K and Mori S (2013) High-throughput neuro-imaging informatics. Front. Neuroinform. 7:31. doi: 10.3389/fninf.2013.00031

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 21 May 2014
doi: 10.3389/fninf.2014.00057

Accumulated source imaging of brain activity with both low and high-frequency neuromagnetic signals

Jing Xiang[1]*, Qian Luo[2], Rupesh Kotecha[1,3], Abraham Korman[1], Fawen Zhang[4], Huan Luo[5], Hisako Fujiwara[1], Nat Hemasilpin[1] and Douglas F. Rose[1]

[1] *Division of Neurology, MEG Center, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA*
[2] *Department of Neurosurgery, Saint Louis University, St. Louis, MO, USA*
[3] *Cleveland Clinic Foundation, Department of Radiation Oncology, Cleveland, OH, USA*
[4] *Department of Communication Sciences and Disorders, University of Cincinnati, Cincinnati, OH, USA*
[5] *State Key Laboratory of Brain and Cognitive Science, Institute of Biophysics, Chinese Academy of Sciences, Beijing, China*

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Stephen Ellis Robinson, National Institutes of Health, USA
Zainab Fatima, Rotman Research Institute, Canada

****Correspondence:***
Jing Xiang, Division of Neurology, Cincinnati Children's Hospital Medical Center, 3333 Burnet Ave., Cincinnati, OH 45229, USA
e-mail: jing.xiang@cchmc.org

Recent studies have revealed the importance of high-frequency brain signals (>70 Hz). One challenge of high-frequency signal analysis is that the size of time-frequency representation of high-frequency brain signals could be larger than 1 terabytes (TB), which is beyond the upper limits of a typical computer workstation's memory (<196 GB). The aim of the present study is to develop a new method to provide greater sensitivity in detecting high-frequency magnetoencephalography (MEG) signals in a single automated and versatile interface, rather than the more traditional, time-intensive visual inspection methods, which may take up to several days. To address the aim, we developed a new method, accumulated source imaging, defined as the volumetric summation of source activity over a period of time. This method analyzes signals in both low- (1∼70 Hz) and high-frequency (70∼200 Hz) ranges at source levels. To extract meaningful information from MEG signals at sensor space, the signals were decomposed to channel-cross-channel matrix (CxC) representing the spatiotemporal patterns of every possible sensor-pair. A new algorithm was developed and tested by calculating the optimal CxC and source location-orientation weights for volumetric source imaging, thereby minimizing multi-source interference and reducing computational cost. The new method was implemented in C/C++ and tested with MEG data recorded from clinical epilepsy patients. The results of experimental data demonstrated that accumulated source imaging could effectively summarize and visualize MEG recordings within 12.7 h by using approximately 10 GB of computer memory. In contrast to the conventional method of visually identifying multi-frequency epileptic activities that traditionally took 2–3 days and used 1–2 TB storage, the new approach can quantify epileptic abnormalities in both low- and high-frequency ranges at source levels, using much less time and computer memory.

Keywords: magnetoencephalography, brain, multi-frequency, high-frequency oscillations, magnetic source imaging

INTRODUCTION

Recent studies have revealed the significance of high-frequency brain signals – such as high-frequency oscillations (HFOs, 90–200 Hz), ripples (80–250 Hz) and fast ripples (250–500 Hz) relative to the conventional lower frequency brain signals (<70 Hz) (Pulvermuller et al., 1997; Guggisberg et al., 2007; Gotman, 2010; Worrell et al., 2012). One of the important motivations behind the study of high-frequency brain signals is their potential clinical applications. HFOs may be important biomarkers of epileptogenicity, a revolutionary finding revealed in recent years (Xiang et al., 2004, 2009a, 2010). Clinical data have revealed that removal of HFO-generating areas lead to improved surgical outcomes (Haegelen et al., 2013). In addition, by using HFOs, it is possible to substantially reduce the extent of cortical resections in epilepsy surgery procedures without compromising seizure control (Weiss et al., 2013). Furthermore, HFOs also play a very important role in many brain disorders (Uhlhaas et al., 2011). For example, schizophrenia is associated with abnormal amplitude and synchrony of high frequency activities (Uhlhaas and Singer, 2013). Of note, the study of high-frequency brain signals may shed light on some of the fundamental mechanisms of neuronal functions and brain disorders.

Numerous challenges exist in the study of high-frequency brain signals with magnetoencephalography (MEG) and electroencephalography (EEG) (Xiang et al., 2004, 2010, 2013; Dalal et al., 2008; Papadelis et al., 2009; Chen et al., 2010; Gotman, 2010; Gummadavelli et al., 2013). First, the size of high sampling rate data can be over 12 terabytes (TB) (Blanco et al., 2011). The size of high sampling rate data can cause a substantial amount of data, posing a challenge for data transfer, storage, archiving, sharing and analysis (Van Essen et al., 2012; Worrell et al., 2012; Zafeiriou and Vargiami, 2012; Zijlmans et al., 2012b). Given the massive amounts of high-sampling rate MEG/EEG data that are collected from patients and research subjects, it is impractical to rely on a visual review of HFOs (Haegelen et al., 2013; Tort et al., 2013; Xiang et al., 2013). Second, in clinical practice, MEG/EEG data are typically analyzed with other neuroimaging data such as invasive recordings, magnetic resonance imaging (MRI) and functional MRI (fMRI). The considerable volume of multi-modal neuroimaging data produced across different communities has posed a daunting challenge to the traditional methods of data sharing, data archiving, data processing, and data interpreting (Van Essen et al., 2012; Worrell et al., 2012; Zafeiriou and Vargiami, 2012; Zijlmans et al., 2012b). Though the multi-modal data enhance our collective understanding of the structure and function of the brain, it is a challenge to handle these varied and heterogeneous datasets. Even with modern computational innovations, there remain technical challenges in data transfer, storage, and analysis of large data sets of more than 12 TB (Brinkmann et al., 2009; Le Van Quyen et al., 2010). Third, the best way to clinically utilize analysis from high-frequency brain signals remains a challenge. While it has been demonstrated that the brain generates signals in wide frequency ranges, there are currently no established criteria for distinguishing physiologic high-frequency signals from pathologic neuromagnetic signals (Worrell et al., 2012; Zijlmans et al., 2012b; Haegelen et al., 2013; Matsumoto et al., 2013; Pail et al., 2013; Srejic et al., 2013; Tort et al., 2013). Although multiple studies with invasive recordings have shown the feasibility and potential clinical importance of detecting HFOs (Jirsch et al., 2006; Engel et al., 2009; Jacobs et al., 2009, 2012; Levesque et al., 2011; Andrade-Valenca et al., 2012; Dumpelmann et al., 2012; Zijlmans et al., 2012b), there is no noninvasive method which can be used for clinical purposes. One remaining important clinical question is whether a noninvasive method can extract and visualize meaningful HFOs from the brain for research and clinical purposes.

This study aimed to resolve the aforementioned challenges associated with large scale high-frequency signal processing by developing novel analysis methodologies and workflows for the MEG data. Since the computer memory limits for a 32 bit and 64 bit operating system are 4 GB and 192 GB (Windows 7, respectively) (http://msdn.microsoft.com/en-us/library/windows/desktop/aa366778(v=vs.85).aspx) and the size of high-frequency brain signals are usually larger than 12 TB (Blanco et al., 2011), one methodological question this study would like to address is whether new algorithms could minimize the use of computer memory and storage. To solve the challenges of analyzing more than 12 TB of both high and low frequency MEG data, we mathematically and experimentally developed a systematic approach to extract meaningful frequency specific and spatiotemporal information from MEG data. Accumulated spectrograms, a technique which maximizes the signal power of the frequency of interest while simultaneously minimizing other frequency contents, provides a novel method of quantifying and visualizing the frequency signatures of brain activity in both low- and high-frequency ranges. Accumulated source imaging, which volumetrically reconstructs source activity in multiple frequency ranges, provides source images for clinicians to analyze epileptic activity at source levels. The central hypothesis of our research is that neuromagnetic brain signals in both low and high frequency ranges could be localized and visualized with accumulated source imaging. The new algorithm calculated optimal channel-cross-channel (CxC) matrices and source location-orientation weights for volumetric source imaging, minimizing multi-source interference and reducing computational cost. To demonstrate the advancements of the new methods in research and clinical settings, MEG data from subjects were obtained, analyzed, and demonstrated in 2D and 3D environments.

MATERIALS AND METHODS

DETECTION OF LOW- AND HIGH-FREQUENCY MEG SIGNALS AT SENSOR LEVELS

Multi-channel MEG data had to be digitized at a high sampling rate because the sampling rate must be at least two times higher than the frequency edge of interest. For the analysis of low-frequency signals, MEG data could be resampled to minimize the use of memory and to improve the computational efficiency. Resampling was done by decimating signals to extract the low frequency data. A low-pass anti-aliasing filter was applied before resampling. The high and low frequency pass-bands depended on the sampling rate and the frequency ranges of interest. In this study, two pass-bands of 1–70 Hz and 70–200 Hz were used. To compute the accumulated spectrogram, filtered MEG data were then segmented into small data segments. The length of the data segments depended on the time window of wavelet-transformation. In this study, we used a 5 s time-window and 600 frequency bins. There was no overlap between segments. Of note, the total length of recorded MEG data did not always match exactly with all of the segments. To solve this problem, data padding (typically, adding zero to make up enough data points for computing) was applied. If there were more than enough data points, the program also allowed for discarding of “extra” data points. The time duration of these segments depended on several factors including the available computer memory, storage spaces and research purposes. Once the time-frequency representations were computed, they were accumulated into one spectrum by adding them together. The “threshold” was used during data accumulating. There were two threshold values: a minimum threshold value and a maximum threshold value. If a time-frequency value was smaller than the minimum threshold value (e.g., background activity) or larger than the maximum threshold value (e.g., artifacts), the value was discarded. Accumulated spectrum is different from an averaged spectrum because the process of accumulating has several parameters: (1) accumulating has two thresholds; and (2) the accumulated data do not have to be averaged. Since the analysis of high-frequency components required high-sampling data, the re-sampling function was critical for low-frequency spectral analysis, which also minimized the use of computer memory. The workflows of data analyses at sensor levels are illustrated in **Figure 1**.

Morlet continuous wavelet transform was used for transforming time-domain data to frequency-domain data (see **Figure 1**). The Morlet wavelet was used because brain activity is nonstationary and the wavelet is better suited for nonstationary data (Ghuman et al., 2011). Wavelet transform can be described by the

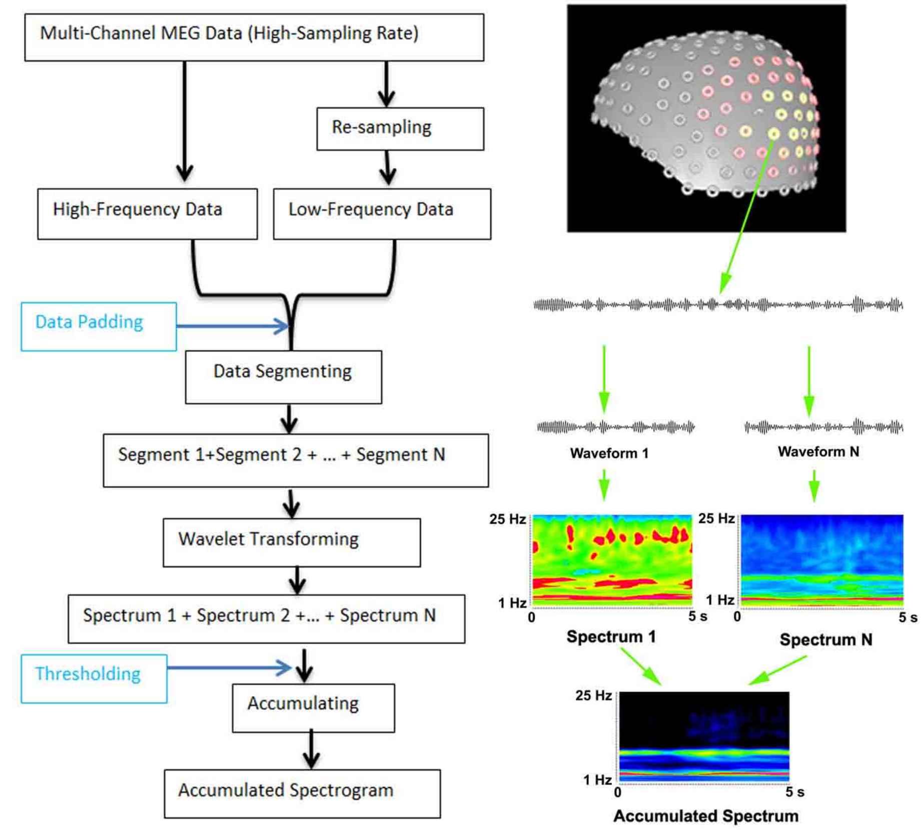


FIGURE 1 | Workflow for computing accumulated spectrogram (left) and the basic principle of computing accumulated spectrogram (right). Since the analysis of high-frequency MEG signals requires high-sampling rate MEG data, MEG data are digitized in a high-frequency range. To improve the performance and optimize the use of computer memory for analyzing both low- and high-frequency MEG signals, the new method can re-sample MEG data dynamically according to the analysis frequency ranges. If the data points of the recorded data are smaller than the minimum data point of wavelet transform in frequency range, the "Data Padding" function can pad some data points so as to meet the requirements of wavelet transform. The "Thresholding" indicates that a spectral value can be rejected or accepted by the accumulated spectrogram according to a threshold value. MEG data recorded are waveforms, which are divided to segments (e.g., "Waveform 1," "Waveform N") to minimize the use of memory for wavelet transform ("Wavelet transform"). In the new method, wavelet transform transfers each segment of waveform data to a spectrum (e.g., "Spectrum 1," "Spectrum N"). Of note, "N" indicates the total number of segments or spectra, which can be theoretically infinitely large. The "+" indicates the process of accumulation, which add all spectra together to produce an accumulated spectrum ("Accumulated Spectrum"). The left view of the sensor distribution of our MEG system is shown on the top right.

following equation:

$$G(t,\ f) = \frac{1}{\sqrt{2\pi f}} e^{\left(\frac{-t^2}{2\sigma^2}\right)} e^{i2\pi ft} \tag{1}$$

In the above formula, t indicates time, f indicates frequency, and σ represents the standard deviation of the Gaussian curve in the time domain. To ensure stability of the wavelet transform, σ is typically larger than $\frac{5}{2\pi f}$. Since the wavelet convolution brings Gaussian temporal blurring with a standard deviation of σ, the effective number of independent samples is $\frac{N-1}{\sqrt{2\pi (f_s\sigma)^2}}$. The f_s represents the sampling frequency of the data and N represents the number of data points.

Since brain activation in a given time-window might occur in different frequency ranges and different frequencies might have different corresponding amplitudes, we used a different sigma value for each frequency to capture the time-frequency changes. Consequently, wavelet Equation (1) can be represented with an alternate representation in Equation (2) as follows:

$$G(t,\ f) = C_\sigma \pi^{-\frac{1}{4}} e^{-\frac{1}{2}t^2} (e^{i\sigma t} - \kappa_\sigma) \tag{2}$$

In the formula, t indicates time and f indicates frequency. Each wavelet transform has its own sigma value. Sigma is the scaling parameter that affects the width of the window. The sigma values are derived from the mother function in wavelet transform by computing the number of small waves for a time-frequency analyses (Ghuman et al., 2011). Sigma values could also be experimentally determined. κ_σ represents the admissibility and C_σ represents a normalized constant. σ represents the standard deviation of the Gaussian curve in the time domain. If signals appeared in the given sensitive time (a small sigma value) and sensitive frequency (a large sigma value) ranges, they would be enhanced.

An accumulated spectrum was defined as the time-frequency summation of a long-time or continuous recording which had a time period at least two times longer than that of the time window of the spectrum. The equation of computing accumulated spectra is given by:

$$Atf(s,\ f) = \sum_{t=1}^{T} \sum_{f=1}^{F} G(t,\ f) \tag{3}$$

In Equation (3), *Atf* represents an accumulated spectrum; *s* indicates the time slice of the spectrum; *f* indicates frequency bands (or bins) of MEG data; *T* indicates total time points of MEG data and *F* indicate the total frequency bands. We defined $s \geq 1$ and $s \leq T/2$. From computer program point of view, the use of computer memory and storage space by Equation (3) depends on the *s*. Even though *T* could be infinitively increasing, the requirements for computer memory and storage remain the same. Consequently, the approach automatically avoided possible "overflow" or "out of space" problems in a long-time or continuous recording for capturing epileptic activity.

An accumulated spectrogram was computed by sequentially transforming each of the segments of waveform data to time-frequency representations using Morlet wavelet algorithm Equation (2) and then accumulating all the spectra together Equation (3). In this procedure, the different spectrograms of individual time segments were mathematically summed together to a single new overall spectrogram. An accumulated spectrogram can reveal brain activity in a consistent frequency range at multiple time windows. It can be considered as a "collective result" for a long-time recording. **Figure 1** demonstrates the basic principles of computing an accumulated spectrogram. An accumulated spectrogram could reveal brain activity in a consistent frequency range while minimizing noise at random frequency ranges (**Figure 1**). Therefore, it could be considered to be a "collective result" of spatial- and frequency locked signals in multiple epochs of MEG data. To identify the frequency profile of the entire brain for a recording, we developed an accumulated global spectrogram. An accumulated global spectrogram was an averaged spectrogram of all accumulated spectrograms from the entire MEG sensor array. The accumulated global spectrogram was the "spatial summation" of the entire MEG sensor array' accumulated spectrograms. Since each sensor was positioned in a distinct location around the brain if there was a subject, an accumulated global spectrogram should represent the magnetic field of the entire brain. The mathematical principles have been described in previous reports (Rau et al., 2002). The neuromagnetic activity at each sensor was visualized with contour maps, which showed small spectrograms at the position of each MEG sensor. The equation of computing global spectrogram is given by:

$$G(s,\ f) = \frac{1}{M} \sum_{m=1}^{M} Aft(s,\ f) \tag{4}$$

In Equation (4), *G* represents the global spectrogram; *Atf* represents an accumulated spectrum of one MEG sensor data; *m* indicates MEG sensor index and *M* indicates the total number of MEG sensors; *s* indicates the time slice of the spectrum; *f* indicates frequency bands (or bins) of MEG data. Since each sensor was positioned in a distinct location around the head (**Figure 1**), the global spectrogram is considered to be a "spatial summation" for each epoch of data (Xiang et al., 2009a).

DETECTION OF LOW- AND HIGH-FREQUENCY MEG SIGNALS AT SOURCE LEVELS

To detect low- and high-frequency neuromagnetic signals at source levels, two computing pipelines were developed. One computing pipeline generated multi-frequency datasets by processing MEG data with filter or wavelet transforms. MEG signals in multi-frequency datasets were in a set of frequency ranges. Of note, the frequency ranges depended on the research tasks and can be predefined. Another computing pipeline performed four tasks: (1) creating a three-dimensional source grid (3D grid), where each grid node represents a possible source; (2) conducting forward solution by calculating lead fields for each source (node) for the entire grid; (3) computing the lead field norm (or magnitude) and ranking the norm for each source for all sensors; (4) producing the node-beam lead field, performing single value decomposition (SVD) and calculating spatial filter weights. The node-beam lead field, which represents a form of sub-space solution, was completed by selecting a group of sensors which had a larger lead field norm. According to our tests, the optimal number of sensors for a node-beam lead field was in a range of 3 to M/3; here M indicates the total number of sensors of a whole cortex MEG system. For example, in our study, the total number of MEG sensor was 275. Thus, the suitable number of sensors that could be used for node-beam lead field was 3–91 (275/3). Of course, all sensors could be used for source scan. A small number of sensors was used in node-beam lead field because high-frequency brain signals were typically very weak and appeared only in a focal group of sensors. The node-beam sensors were also used to generate beam sensor MEG datasets so that the sensors in forward solution matched with measured magnetic signals. The final step was to compute source moments and to generate source data. Additional components were optional (red lines, which will be discussed in following sections). The main workflow for localizing both low- and high-frequency MEG signals is shown in **Figure 2**.

Differing from the conventional volumetric source imaging or distributed source map, each grid node consisted of multiple data items including the strength and frequency of the source activity (**Figure 2**). Building on previous reports (Mosher and Leahy, 1998; Vrba and Robinson, 2001; De Gooijer-Van De Groep, 2013), the mathematic relationship between measured MEG data and source activity can be expressed as following equation:

$$B = LQ + N \tag{5}$$

In Equation (4), *B* represents the MEG data; *L* represents the lead field, *Q* represents the source strength, and *N* represents the noise. For a given MEG dataset, *B* is known and *L* can be computed for each node with a forward solution. The forward solution in

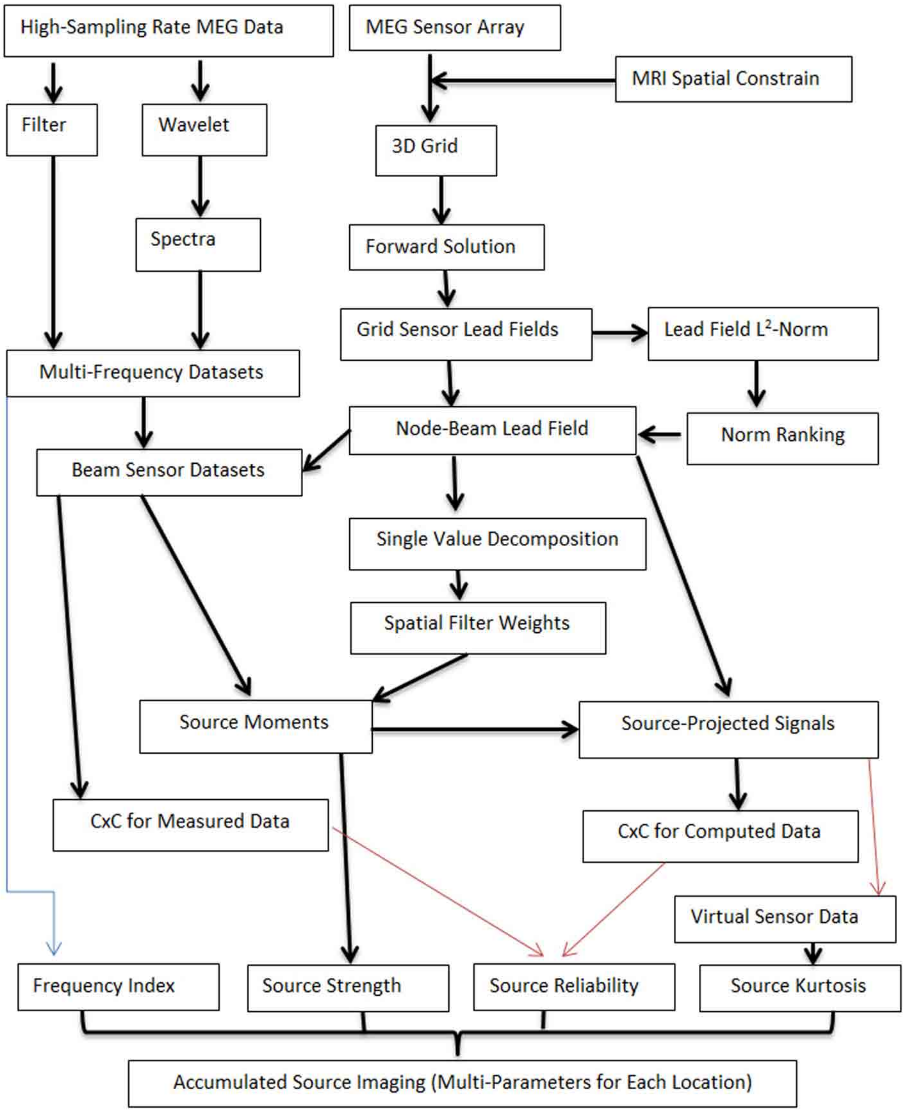


FIGURE 2 | Workflow for computing accumulated source images. The workflow includes two main computing pipelines. One computing pipeline processes MEG data with filter or wavelet transforms so as to generate multi-frequency datasets. MEG signals in multi-frequency datasets are in a set of frequency ranges. Another computing pipeline works on several tasks, which included the creation of a three-dimensional source grid (3D grid), performing forward solution by calculating lead fields, ranking the norm for each source for all sensors, and performing SVD. The node-beam lead field is completed by selecting a group of sensors which have a larger lead field norm (or weights). Of note, each location in accumulated source imaging can have multiple parameters (e.g., "Frequency Index," "Source Strength"). Some processes are optional (red lines) and additional parameters can also be added to the workflow.

this study was computed according to Sarvas' formula for outside hemispherical conductors in Cartesian coordinates (Sarvas, 1987).

The determination of source strength and orientation of Q has been a challenge as discussed in many previous reports (Mosher et al., 1999; Huang et al., 2004; Robinson, 2004; De Munck and Bijma, 2009; Ou et al., 2009). According to our tests, the determination of MEG data in both low- and high-frequency ranges with conventional beamforming required considerable time and computing power to decompose MEG sensor data to subspaces because the data in both low- and high-frequency ranges had more data points as compared with the previous reports typically focusing on a single frequency range. However, for a given MEG data set in multiple frequency ranges in a limited time window (2 min in this study), the positions of sensor array and the 3D source grid were fixed; consequently, lead fields could be computed once and then used for both low and high-frequency ranges. Under these assumptions, we propose using SVD to decompose the lead field as following:

$$L = USV^T \tag{6}$$

Where $U \in R^{mxm}$ is an orthogonal (unitary in the complex case) matrix. The columns of U are the left singular vectors of L. $V \in R^{mxm}$ is an orthogonal (unitary in the complex case) matrix. The columns of V are right singular vectors of L. $S = diag(\sigma_1,\ \sigma_2, \ldots \sigma_p)$ is an $M \times N$ diagonal matrix with $p = \min(m,\ n)$ and $\sigma_1,\ \sigma_2, \ldots \sigma_p$ are the singular values of L. M indicates the number of sensors and N indicates the number of source orientations. For a single source, $p = 3$. The Moore-Penrose pseudo inverse of L is given by:

$$L^+ = VS^+U^T \tag{7}$$

Where S^+ is a diagonal formed with the multiplicative inverses of the nonzero singular values of L placed on the diagonal. Assuming there was no noise ($N = 0$), the measured MEG data, B, can be described by the following equations:

$$B = LQ = USV^TQ \tag{8}$$

$$Q = BL^{-1} \tag{9}$$

By replacing L^{-1} in Equation (9) with L in Equation (8), the estimated moment, $\vec{Q}$, can be computed with a SVD back substitution as described in the following equation:

$$\vec{Q} = BVS^+U^T \tag{10}$$

Of note, L^+, pseudo inverse of L, could be computed once and used for the analysis of data in all frequency ranges, which makes the computation of source strength and probability more efficient. In addition, once the $\vec{Q}$ is determined, virtual sensor spectrograms can be also computed with $\vec{Q}$ for each frequency range and time window.

$$V(t,\ f) = \sum_{t=1}^{T}\sum_{f=1}^{F} ||\vec{Q}||_2 (TF)^{-1} \tag{11}$$

In Equation (11), V represents the computed virtual sensor spectral data. The t and T indicate time slice and total number of time windows, respectively. The f and F indicate frequency band and total number of frequency bands, respectively. Magnetic signals generated by $\vec{Q}$ can be computed with the follow equation:

$$X_{cmp} = L\vec{Q} \tag{12}$$

where $Xcmp$ represents computed magnetic signals at individual sensors from source $\vec{Q}$. We used $Xmea$ to represent the measured magnetic signals at individual sensors, which were different from B in Equation (3), which represents MEG data in general.

RELIABILITY ASSESSMENT OF SOURCE ACTIVITY AT LOW- AND HIGH-FREQUENCY RANGES

To minimize the "ill-posed" inverse problem in MEG, the theory that a given MEG sensor data pattern may have an infinite number of possible "correct" answers (Hamalainen and Sarvas, 1987; Sarvas, 1987), we developed a channel-cross-channel (CxC) function to analyze the spatial pattern of MEG signals. Building on the use of covariance matrix for MEG beamforming in our previous studies (Kotecha et al., 2009; Gummadavelli et al., 2013), we applied a subtraction operation to all possible channel-pairs to generate a matrix which described the spatial gradient of magnetic signals among the sensors. Mathematically, each entry outside of the main diagonal in a CxC matrix represents the difference of a channel-pair. The diagonal entries represent the values of the corresponding sensors. To assess the reliability of source activity, the similarity of the measured MEG signal ($Xmea$) and the computed MEG signals ($Xcmp$) were statistically analyzed with the CxC matrix by computing the covariance and correlation factors with the following formulas:

$$C\left(x_{mea},\ x_{cmp}\right) = \frac{\sum_{i=1}^{K}(x_{mea}i - \overline{x_{mea}})(x_{cmp}i - \overline{x_{cmp}})}{N-1} \tag{13}$$

$$R\left(x_{mea},\ x_{cmp}\right) = \frac{C(x_{mea},\ x_{cmp})}{Sx_{mea}Sx_{cmp}} \tag{14}$$

Where $C\left(x_{mea},\ x_{cmp}\right)$ indicates the covariance and $R\left(x_{mea},\ x_{cmp}\right)$ indicates the correlation in the CxC matrices. The x_{mea} and x_{cmp} indicate signals in two channels which were paired for computing CxC. $\overline{x_{mea}}$ and $\overline{x_{cmp}}$ represent the mean of the signals in the measured and computed datasets, respectively. Sx_{mea} and Sx_{cmp} indicate the standard deviation of the signals in the two datasets, respectively. K indicates the number of sensors used for source estimation, which was smaller or equal to the total number of measuring sensors. To statistically determine the spatial correlations for each node in the 3D grid, t-values were computed for all sources.

$$Tp = R\sqrt{\frac{K-2}{1-R^2}} \tag{15}$$

In Equation (15), Tp is the t-value of a source; R indicates the correlation of the measured and computed MEG signals for the source; K indicates the number of sensors related to the source.

A careful observation of Equation (13) could find that x_{cmp} is similar to the weights of the conventional beamforming because x_{cmp} represents signals from a predefined location and estimated source orientation. Similar to the conventional beamforming, the use of x_{cmp} could maximize signals from the source and minimize environmental noise and signals from other locations. For the analyses of multi-frequency signals, the location-orientation weights were computed from the optimal CxC matrix for each frequency. Thus, the source orientation was independent of frequency and only dependent on the orientation of the cortical normal vector. In other words, the solutions are approximations; the orientation portion was frequency independent.

Building on previous reports that the spectral signatures of low- and high-frequency signals at source levels can be measured with the combination of accumulated spectrogram and virtual sensors (Xiang et al., 2004, 2009a,b; Xiang and Xiao, 2009), the present study developed accumulated source imaging (**Figure 2**). With this technique, an accumulated source image was generated by accumulating all the source data computed for each location and each frequency band from the entire epoch of the MEG data. Of note, the computing of accumulated source images maintained spatial- and frequency-locked signals and minimized signals in random-space and frequency.

MAGNETIC SOURCE IMAGING WITH MULTI-PARAMETERS PER LOCATION (MPPL)

This study moved one step further by developing magnetic source images with multi-values per location or MPPL. Specifically, each location has multi-parameters: (1) the first parameter describes the frequency range, which is represented with a frequency index for minimizing the use of computer memory and storage spaces; (2) the second parameter describes the strength of source activity; (3) the third parameter describes the reliability of the source; (4) the fourth parameter describes the Kurtosis or "peakedness" of source activity. The frequency index was directly obtained from the processed MEG data (the values of high-pass and low-pass filters or the frequency index in time-frequency representation). The strength of source activity was the source moment computed with Equation (10). The reliability could be computed with Equations (14) or (15). Building on previous report (Robinson et al., 2004), the Kurtosis was computed with following equation.

$$K = \frac{\sum_{t=1}^{T} (q(t) - u)^4}{T\sigma_t^4} - 3 \tag{16}$$

Where T is the length of source data t in a time window, which has a mean of u and a standard deviation of σ. K represents the kurtosis values and is stored in parameter 4 in accumulated source imaging.

As shown in **Figure 2**, the analyses of MEG signals at both low- and high-frequency ranges generated more than one value for each location or each node of the 3D grid (e.g., strength, reliability and frequency of source activity). Notably, conventional magnetic source imaging, which encodes one value for one location or voxel, cannot represent the source data computed with the developed methods. The main differences between the new methods and existing methods are summarized in **Table 1**.

SOURCE LOCALIZATION WITH ACCUMULATING

Accumulated source imaging was defined as the volumetric summation of source activity over a period of time which was at least two times longer than that of the time window of the source image. Of note, accumulated source imaging could have more than 1 time slices to reveal the fluctuation of source activity in space and time. Accumulated source imaging can be described as the following equation:

$$Asi(r,\ s) = \sum_{t=1}^{t=n} Q(r,\ t) \tag{17}$$

In Equation (17), Asi represents accumulated source strength at location r; s indicates the time slice; t indicates time point of MEG data; n indicates total time points of MEG data and Q indicate the source activity at source r and at time point t. We defined that $s \geq 1$ and $s \leq n/2$. From a computer program point of view, the use of computer memory and storage space by Equation (12) is dependent on the s for a fixed source imaging configuration (e.g., spatial resolution and dimension). Even though n could be infinitely increasing, the requirements for computer memory and storage remain the same. Consequently, the approach automatically avoided possible "overflow" or "out of space" problems in a long-time or continuous recording for capturing epileptic activity such as spikes. Since accumulated source imaging accumulates the results of source data, it is different from previous reports which compute a covariance matrix or kurtosis of sensor data for a long-time recording. Specifically, using a covariance matrix or kurtosis computed with sensor data for a long-time recording for source localized is based on the assumption that the source was stationary during the long-time recording. Our approach, on the other hand, did not make this assumption. Therefore, our approach has the capability to detect both stationary and nonstationary source activity.

MEG EXPERIMENTS, MRI SCAN AND INTRACRANIAL RECORDINGS

Participants

Ten healthy children (5 girls; 5 boys; age: 6–18 years; mean age: 12.8 years) were recruited for this study. Inclusion criteria were: (1) healthy without a history of neurological disorders or brain injuries; (2) age-appropriate functions including hearing, vision, and hand movement; (3) head movement during MEG recording was less than 5 mm. Ten pediatric patients (5 girls; 5 boys; age: 6–18 years; mean age: 12.7 years) with clinically diagnosed epilepsy were retrospectively studied. Patient inclusion criteria were: (1) clinically diagnosed epilepsy; (2) head movement during MEG recording was less than 5 mm; and (3) epileptic foci

Table 1 | Differences between accumulated source imaging (ASI) and similar methods.

	ASI	DM	SAM	SAM(g2)	BF	MN	MUSIC
Optimized for localizing HFOs	Yes	No	No	No	No	No	No
Handle large dataset	Yes	No	No	No	No	No	No
Handle multi-frequency signals	Yes	No	No	No	No	No	No
Multi-parameter per location	Yes	No	No	No	No	No	No
Volumetric source scan	Yes	No	Yes	Yes	Maybe	Yes	Yes
Detect dynamic sources	Yes	Yes	No	No	No	Yes	Yes
Detect stationary sources	Yes	No	Yes	Yes	Yes	No	No
Detect correlated sources	Yes	Yes	No	No	No	Yes	Yes
Noise suppression	Yes	No	Yes	Yes	Yes	No	Yes

ASI, accumulated source imaging; DM, dipole modeling (dipole fitting); SAM, synthetic aperture magnetometry; SAM (g2), SAM excess kurtosis (g2); BM, conventional beamforming; MN, minimum-norm; MUSIC, multiple signal classification.

were confirmed with electrocorticography (ECoG) and/or neuroimaging data. Exclusion criteria were: (1) inability to remain still; and (2) presence of an implant such as a cochlear implant device, a pacemaker, or a neuro-stimulator containing electrical circuitry, generating magnetic signals, or having other metal that could produce visible magnetic noise (>6 pT) in the MEG data. Written consent, formally approved by the Institutional Review Board (IRB) at Cincinnati Children's Hospital Medical Center (CCHMC) and Nanjing Brain Hospital, was obtained from each healthy participant prior to testing. This study was approved by IRB at CCHMC.

MEG recordings

MEG signals were recorded in a magnetically shielded room (MSR) using a whole head CTF 275-Channel MEG system (VSM MedTech Systems Inc., Coquitlam, BC, Canada) in the MEG Center at CCHMC. Before data acquisition commenced, three electromagnetic coils were attached to the nasion, left and right pre-auricular points of each subject. These three coils were subsequently activated at different frequencies for measuring each subject's head position relative to the MEG sensors. Each subject lay comfortably in the supine position, his or her arms resting on either side, during the entire procedure. MEG data were recorded at a sampling rate of 4000 Hz. Continuous MEG recordings were completed for an epoch of 2 min. To ensure the reproducibility, at least two epochs were recorded for each subject. All MEG data were recorded with a noise cancellation of third order gradients and without on-line filtering. To identify system and environmental noise, we routinely recorded one MEG dataset without a subject immediately prior to the experiment.

MRI scan

Three-dimensional magnetic resonance imaging (MRI) was obtained using a 3-T Philips Achieva scanner (Philips Healthcare, Andover, MA). Three fiduciary marks were placed in identical locations to the positions of the 3 coils used in the MEG recordings with the aid of digital photographs to allow for an accurate co-registration of the 2 data sets. Subsequently, all anatomic landmarks were made identifiable in the MRIs.

Similar to previous reports (Xiang et al., 2009a), clinical intracranial electrocorticography (ECoG) data were retrospectively analyzed with the MEG results. Of the 10 patients, the 8 patients reported here had implantation of subdural electrodes and CCTV/EEG (VEEG) monitoring according to standard protocol at our hospital. Digital photos were taken before and during the operation to record the placements of the electrodes.

IMPLEMENTATION OF THE ALGORITHMS

The aforementioned method for reconstruction of brain activity was implemented in MEG Processor with C/C++ on Windows platform (Xiang et al., 2010; Gummadavelli et al., 2013). MEG Processor was driven by its Windows interface. From the user perspective, its organization is contextual rather than linear: the multiple features from the software were not listed in long menus, they were accessible only when needed and were typically suggested within contextual popup menus or specific interface windows. This structure provided faster and easier access to requested functions.

DATA ANALYSES

MEG data were visually inspected for artifacts. MEG waveforms with identifiable artifacts (amplitude >6 pT) were excluded from data analyses. Similar to previous reports (Xiang et al., 2009a), accumulated spectrograms, global spectrograms and spectral contour maps for all subjects were computed and analyzed. Before reconstructing brain activity for human MEG data, the head was modeled as a homogenous conducting sphere in order to account for volume-conducted return currents. The sphere model used in this study was a multiple local-sphere model, where each sphere (one per MEG sensor) was fit to a small patch of the head model (directly under the sensor) in order to better model the local return currents (Huang et al., 1999). The conducting boundary was defined with individual MRI, which was the inner skull. In other words, the best-fit sphere was fit to the scalp. From this head model, a whole-brain, subject-specific lead field was computed and used for magnetic source reconstruction. Accumulated source imaging and conventional beamforming (Vrba and Robinson, 2001) were implemented in MEG Processor for source estimation (Kotecha et al., 2009; Chen et al., 2010; Gummadavelli et al., 2013). CTF software package (VSM MedTech Systems Inc., Coquitlam, BC, Canada) was used to perform dipole fit analyses (Robinson et al., 2004; Kirsch et al., 2006). We used MNE (Gramfort et al., 2014) and Brainstorm (Tadel et al., 2011) to perform source estimation with Minimum-norm and multiple signal classification (MUSIC) algorithms, respectively.

To quantify the results, electrocorticography (ECoG) was used as the "gold standard" for defining epileptic zones. MEG sources were overlapped onto individual MRI data. Cerebral landmarks including the central sulcus, Sylvian fissure and the somatosensory cortex were used to define specific anatomical cortical brain regions (Agirre-Arrizubieta et al., 2009). The brain regions were the central, parietal, and occipital lobes. The frontal lobe was divided in the frontal superior, medial, inferior, and fronto-orbital regions; the temporal lobe into the lateral and mesial regions, the latter comprising the amygdala, the hippocampus, the parahippocampal gyrus, and the temporal-basal area. The inter-hemispheric region consisted of the mesial surface of the frontal, parietal, and occipital lobes (De Gooijer-Van De Groep, 2013). Similar to previous reports (Agirre-Arrizubieta et al., 2009; De Gooijer-Van De Groep, 2013), the concordance between MEG sources and ECoG was measured by determining if the interictal ECoG and MEG source locations were anatomically matched in the brain regions. We defined the sensitivity and specificity of the methods as followings.

$$Sensitivity = \frac{TP}{TP+FN} \tag{18}$$

$$Specificity = \frac{TN}{TN+FP} \tag{19}$$

Where TP represents the number of true positive (both MEG and ECoG showed epileptic foci); FN indicates the number of false negative (ECoG showed epileptic foci while MEG showed no epileptic focus); TN represents the number of true negative (both MEG method and ECoG showed no epileptic foci); FP represents number of false positive (MEG showed epileptic foci, but ECoG showed no epileptic focus).

STATISTICAL ANALYSIS

The comparisons of spectral and source data for epilepsy subjects and controls were performed with paired Student T-tests. The odds ratios of activity in brain areas in epilepsy subjects other than the areas identified in control groups for each frequency band were analyzed with Fisher's exact tests. Significance was accepted at the level of $p < 0.05$ for one comparison. Since multiple frequency bands and more than one source were analyzed, Bonferroni multiple comparison corrections were applied. Specifically, if multiple comparisons were to be taken into account then the significance level for any one of these comparisons was reduced from 0.05 to 0.05/parameter (e.g., for 9 frequency bands, $p < 0.005$).

RESULTS

The size of 2 min MEG data digitized at a sampling rate of 4000 Hz (CTF MEG system, 275 sensors) was 0.597 GB. For time-frequency analyses, if the frequency bin of time-frequency transform was 600, the size of the time-frequency representation of 0.597 GB waveform data were 358 GB (600 × 0.597). When we computed the CxC data with time frequency data, the size of time-frequency based covariance matrices were 128164 GB (358 × 358 GB), which was approximately 125 TB. Of note, the source data computed from the time-frequency data would also be larger (>125 TB). Since the physical memory limit for windows 7 (64 bits, professional version) was 192 GB, the spectral data computed with the conventional time-frequency analysis method could not be stored in our Windows workstations because as it clearly exceeded the upper limits of the operating system. Alternatively, with accumulated spectrogram, we were able to limit the size of the spectrogram to 3 GB. Noticeably, the size and time required for computing an accumulated spectrogram mainly depended on the dimension of the accumulated spectrogram (number of frequency bins and time slices) and frequency ranges which could be adjusted by users. For an accumulated spectrogram with a dimension of 600 × 600 (600 frequency bins, 600 time slices) for a 2 min recording (sampling rate 4000 Hz), it took approximately 8.1 ± 0.03 h for data in 70–200 Hz, 1.3 ± 0.002 h for data in 1–70 Hz. Of note, the processing time would also depend on the speed of CPU and GPU, the number of programs running, the optimization of software compiling. In this study, we used two CPU (Intel Xeon, E4506, 2.13 Hz, each CPU has four cores). If GPU was used, the times were shortened to 42.5 ± 0.31 min for data in 70–200 Hz and 12.2 ± 0.009 min for 1–70 Hz, respectively. GPU could significantly shorten the computing time ($p < 0.0001$). Examples of accumulated spectrograms are shown in **Figures 3–6**. To identify high-frequency signals in multiple frequency bands with visual inspection, it took 2–3 days for a neurologist with 8 years of EEG/MEG experience.

The processing time for source scan with the conventional dynamic multi-dipole modeling (finding the 13 dipole for each time-slice) in multi-frequency ranges for recording at a sampling rate of 4000 Hz took 92.3 ± 0.4 h. However, our accumulated source imaging, which automatically scanned the entire brain for the same dataset took 12.7 ± 0.4 h. Of note, the approach was approximately 7.6 times faster than the conventional approach ($p < 0.0001$). If GPU was used, the time could be significantly shortened to approximately 6.3 ± 0.1 h. However, the use of GPU slowed down the user responses in our tests.

The global spectrograms of MEG datasets recorded from three conditions (no subject, healthy subjects and epilepsy subjects) showed that the epilepsy subjects had significantly increased spectral power. **Figure 3** shows an example of global spectrograms in the three conditions. We noted that accumulated spectrograms revealed a clear alpha activity (approximately 8–12 Hz) in all healthy subjects (10/10, 100%) (**Figure 4**). Out of the 10 epilepsy patients, 9 patients showed increased spectral power in 70–200 Hz (9/10, 90%). Further analyses revealed that increased spectral power were around 106, 140, and 168 Hz in epilepsy patients (**Figure 5**). **Figure 6** shows the spatial distributions of accumulated spectrograms in spectral contour maps.

Accumulated source imaging revealed focal increase of spectral power (**Figure 7**). Accumulated source imaging in low frequency ranges revealed that brain activities in 8–12 Hz (alpha) were localized to the occipital cortex in all the healthy subjects (10/10, 100%). However, alpha activity were localized to the occipital

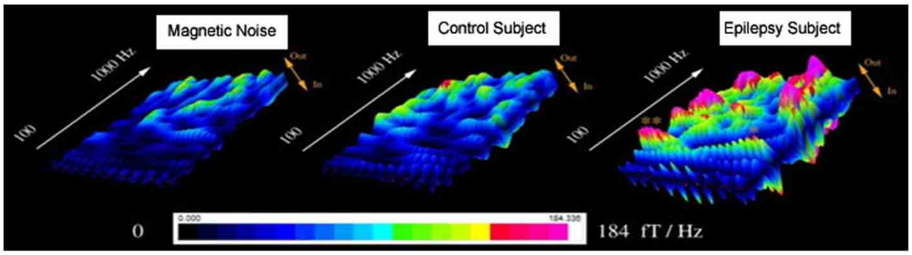


FIGURE 3 | Accumulated global spectrograms in three conditions. "Magnetic Noise" was computed with MEG data recorded without subjects. "Control Subject" was computed with MEG data recorded from a healthy child. "Epilepsy Subject" was computed with MEG data recorded from a child with epilepsy between seizures (interictal). The sampling rate of all MEG recordings was 6000 Hz. An accumulated global spectrogram represents the "spatial summation" of the entire MEG sensor array accumulated spectrograms. The three spectrograms show that the epilepsy subject has elevated spectral power as compared to the control subject.

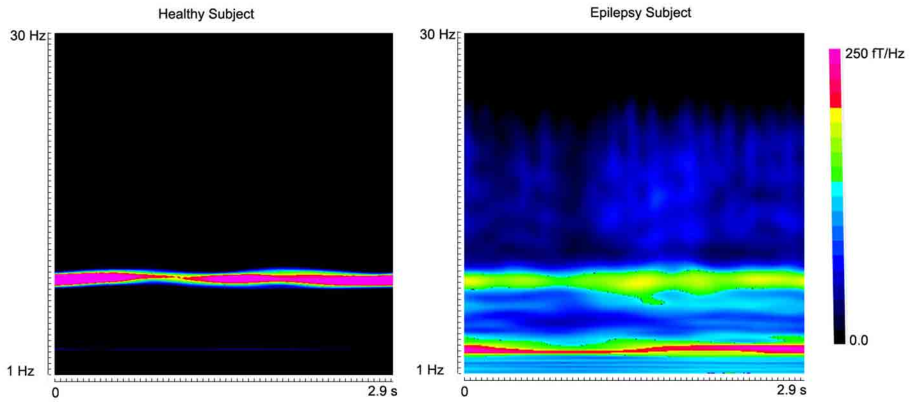


FIGURE 4 | Accumulated spectrograms show the well-known alpha activity in a healthy subject and an epilepsy subject. Noticeably, healthy subject ("Healthy Subject") has a clear activity around 8–12 Hz (alpha activity). However, the epilepsy subject ("Epilepsy Subject") has incrased activity in 2–4 Hz (low-frequency activity). The color bar shows the color coding of spectral power.

cortex in five epilepsy patients (5/10, 50%) and in nonoccipital cortices in other five patients (see **Figure 8** for example). The five patients all had strong epileptic activity, which overshadowed and/or interrupted alpha activity. We noted that the increased spectral power at source levels varied among epilepsy patients.

Accumulated source imaging showed that 9 out of the 10 epilepsy patients (9/10, 90%) had increased focal spectral power in high-frequency ranges at source levels. The epileptic areas localized by accumulated source imaging were concordant with clinical data. **Figure 9** shows an example of epileptic foci volumetrically localized with high-frequency accumulated source imaging (70–200 Hz). The sensitivity and specificity of all subjects are shown in **Table 2**.

DISCUSSION

The present study demonstrated an approach for detecting both low- and high-frequency neuromagnetic signals by integrating time-frequency transform, source localization, accumulation and MPPL algorithms into a comprehensive and systematic processing package. The strengths of our methodologies are reflected by the major features of our signal processing algorithms as well as their abilities to resolve the difficulties associated with the large data volume, multi-modality data and its clinical applicability.

FEATURES OF OUR SIGNAL PROCESSING ALGORITHMS

One of the unique features in our wavelet transform algorithm was that the sigma value (number of waves) could be dynamically changed so as to match the neurophysiological patterns. For example, neuromagnetic signals from a brain area may appear in multiple frequency ranges but in a similar time window. The conventional wavelet algorithm typically gives a wide time-window for low signals and a narrow time-window for high-frequency signals, which is not well-suited for analysis of brain activity. The improved wavelet transform algorithm in the present study could solve this problem by dynamically changing the sigma values so as to adjust the time-window for a better analysis of brain activity.

Accumulating algorithms in the computing of accumulated spectrograms provides a novel method for handling the large datasets obtained when analyzing both low and high-frequency MEG signals. Integration of time-frequency analysis and accumulation into a workflow system is a novel neuroimaging data processing algorithm technique that can summarize and visualize high-frequency signals with a few images.

CxC matrices and functions are critical to the study of neural HFOs. Since high-frequency signals are typically obscured by low-frequency signals (Xiang et al., 2009a), time-frequency representations were normalized according to the magnitude of each frequency bin across all MEG sensors to ensure that all frequency bins contributed equally to the source reconstruction. The time-frequency matrix allows for matrix operations such as subtraction of control state MEG signals from the activation state MEG signals, whose purpose is to increase signal-to-noise ratio (SNR) or to maximize the signal power at a peak frequency (or a frequency of interest) while simultaneously minimizing it at the neighboring surrounding frequency bins. CxC matrices based on the time-frequency data provide unique spatial patterns and gradients of magnetic fields for determining high-frequency sources.

The major differences between our technique and existing methods of volumetric imaging such as beamforming, minimum-norm are the features of accumulation and MPPL, which are more than a source localization algorithm. To our knowledge, none of the existing methods have the features of accumulation and MPPL. It is necessary to point it out that, some existing methods have internally fixed frequency ranges (e.g., 20–70 Hz) (Robinson et al., 2004), which could not be directly compared in our tests because our method was designed to analyze both low- and high-frequency signals (multi-frequencies). Of note, each method has its strengths and weaknesses (De Gooijer-Van De Groep, 2013). According to on our clinical experience, the unique features of the method are clinically important and necessary. For example, the conventional beamforming could also been used to detect multi-frequency signals (Vrba and Robinson, 2001). However, the conventional beamforming is based on covariance matrices, which are computed from data in long time-windows (if the time-windows are short, the sizes of the source data would be a problem). The process assumes that the brain

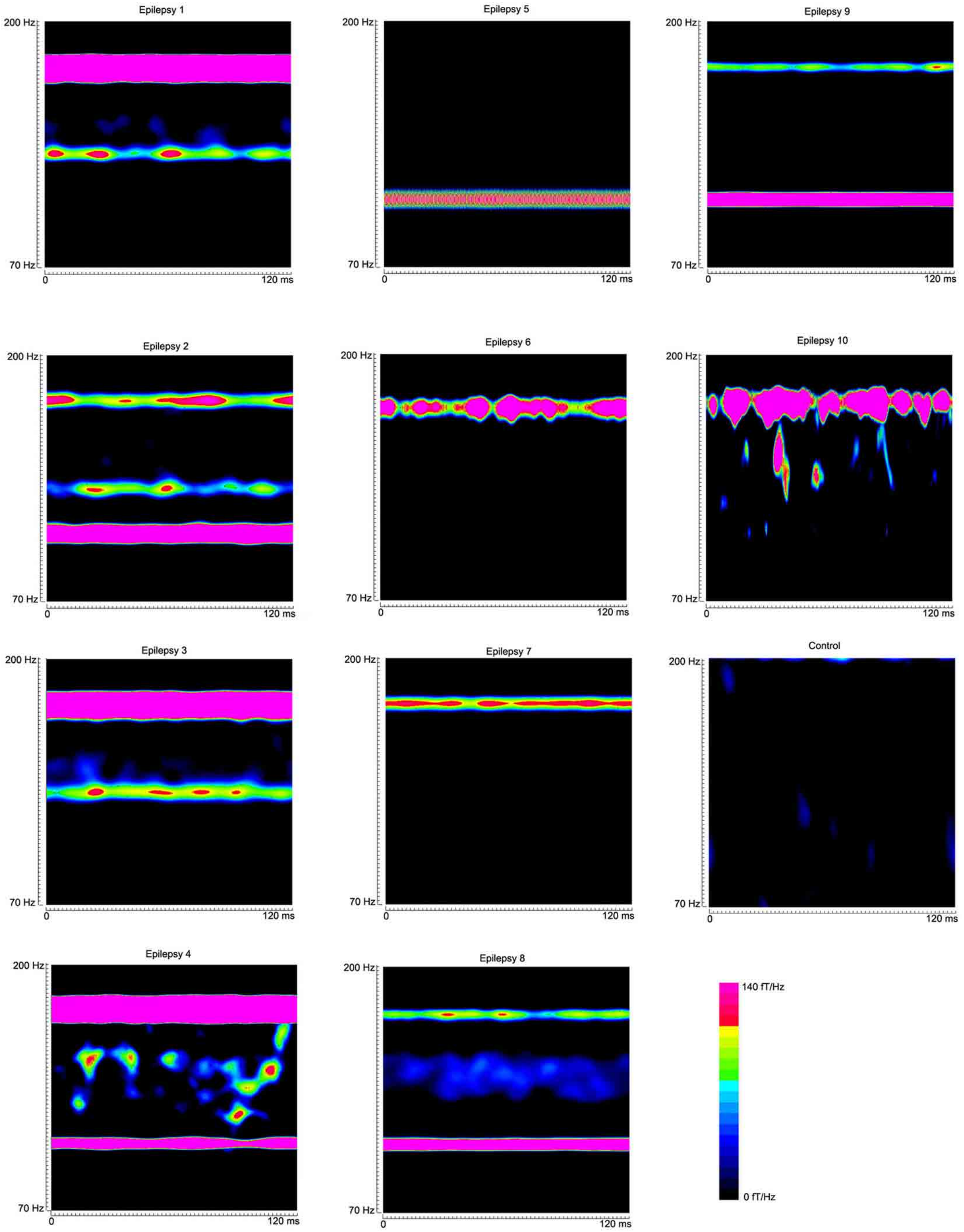


FIGURE 5 | Global accumulated spectrograms from 10 epilepsy subjects and 1 healthy subject show the main frequency components of neuromagnetic signals in 70–200 Hz in epilepsy patients. Noticeably, the activity patterns vary across patients. The color bar shows the color coding of spectral power for all the global accumulated spectrograms.

activity is stationary in the time-window (Vrba and Robinson, 2001), which may be not true for real epileptic activity (Zijlmans et al., 2012b). By using accumulation algorithms, our approach does not making any assumption about the stationarity of the sources. Another example is SAM (g2). SAM (g2) is an outstanding method for detecting excess kurtosis (Kirsch et al., 2006). SAM (g2) is designed for detecting rare events (spikiness activity). It has been shown that combining SAM(g2) and other methods such as MUSIC gives the best clinical results (De Gooijer-Van De Groep, 2013). The development of MPPL in our method enables us to implement both kurtosis and other algorithms by using multiple parameters during source analyses. Consequently, both rare events (kurtosis) and common events (frequent spikes) could be detected by our methods.

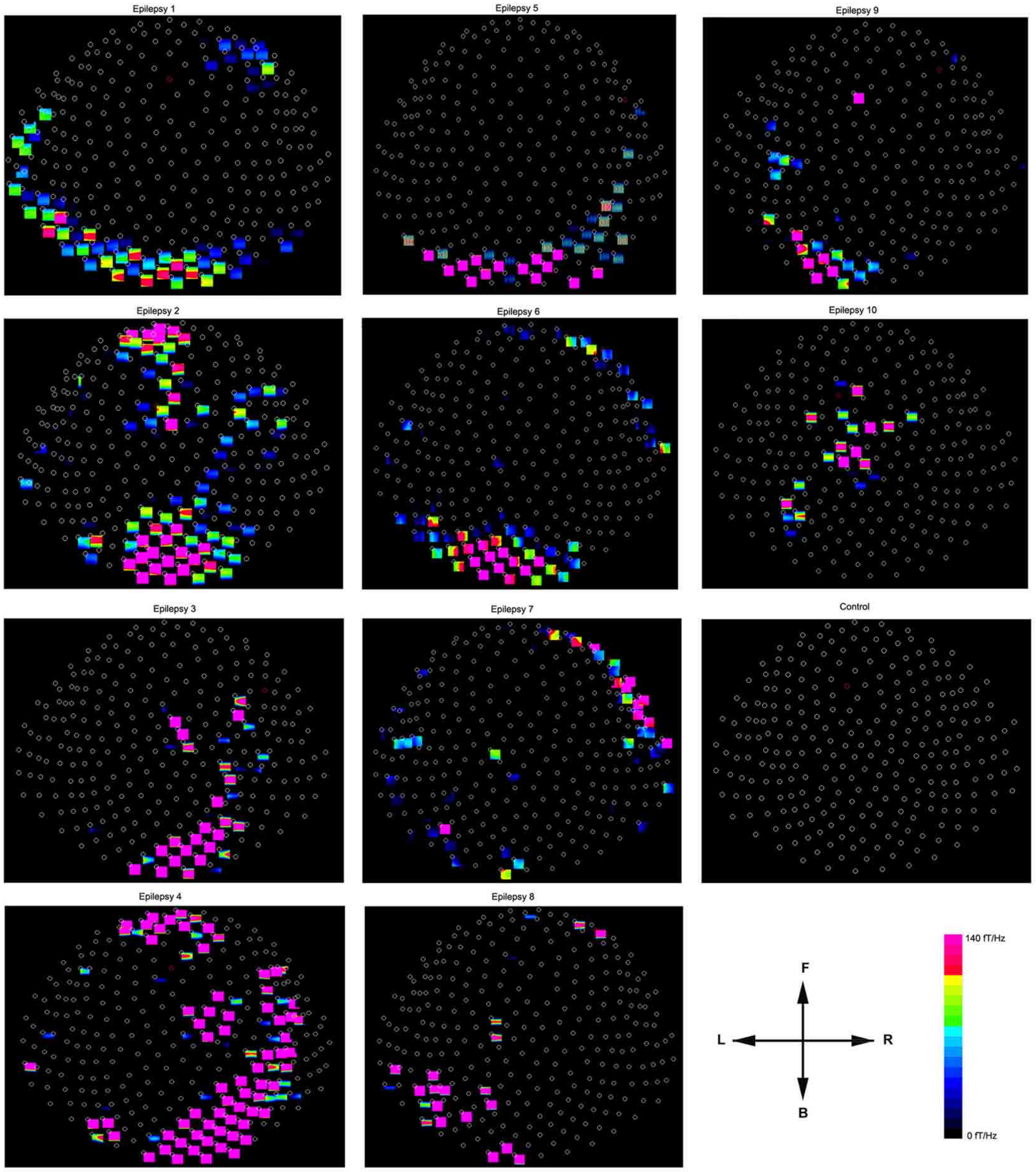


FIGURE 6 | Accumulated spectral contour maps from 10 epilepsy subjects and 1 healthy subject show the spatial distributions. Noticeably, the spectral distribution varies across patients. All the contour maps have the same orientation defined by the arrows: the "L" indicates the left side of the head and the "R" indicates the right side of the head. The "F" indicates that the upper part of the contour map represents the frontal region of the head; the "B" indicates that the lower part of the contour map represents the posterior region of the head. Each small circle represents one MEG sensor. The color bar shows the color coding of spectral power for all the contour maps.

SEVERAL MAJOR CHALLENGES RESOLVED WITH THE CURRENT METHODOLOGIES

Data volume challenge

Our results are consistent with previous reports (Blanco et al., 2011), the size of high sampling rate MEG/EEG data could be in the magnitude of TB (>12 TB). This was particularly true for multi-frequency spectral data (>125 TB). However, by using accumulating techniques, we were able to minimize the size of MEG data to less than 10 GB without losing the high-frequency information. Although accumulated spectrogram was utilized with MEG in the present study, the same technology can also be used in the analysis of EEG and intracranial EEG. In current clinical research, the detection and labeling of interictal and ictal epileptiform activity in intracranial EEG recordings is performed by expert review. This manual method has been known to be associated with a poor inter-reviewer reliability (Benbadis et al., 2009). In addition, manual review is not feasible for large data sets because it is very time consuming and labor-intensive (Restuccia et al., 2011; Andrade-Valenca et al., 2012; Dumpelmann et al., 2012; Jacobs et al., 2012; Zijlmans et al., 2012a; Haegelen et al.,

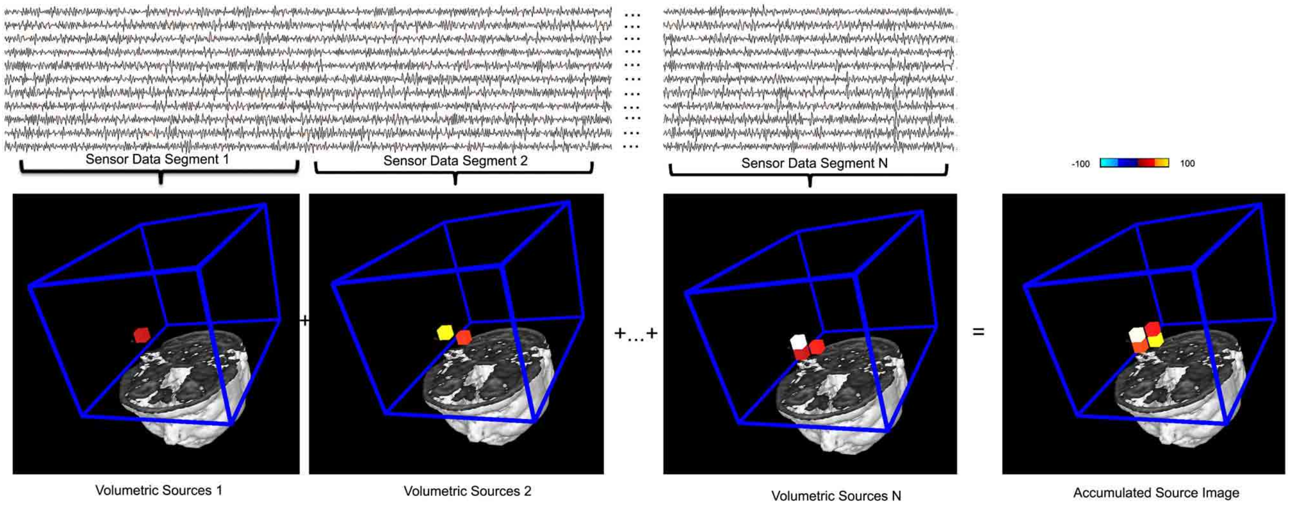


FIGURE 7 | An illustration of the basic principle of accumulated source imaging. The top waveforms show MEG data at sensor levels. The bottom images show individual structural magnetic resonance image and the region of interest (ROI, blue lines) for source scanning. MEG sensor data are firstly divided into small segments (e.g., "Sensor Data Segment 1," "Sensor Data Segment 2," "Sensor Data Segment N"). Volumetric sources are then produced by scanning the entire ROI with each segment of sensor data. The red, yellow and white small cubes indicate the sources (or voxels) identified. For illustration purposes, a very low resolution (12 millimeter) spatial resolution was used. An accumulated source image is generated by spatially adding all volumetric sources together. Of note, only sources reach certain thresholds (in this case, 75%) are added to accumulated source images, which differentiate this accumulating process from averaging. The color bar indicates the color coding of the source strength.

2013; Stacey et al., 2013). Alternatively, accumulated source imaging can automatically analyze large data sets and provide images for experts to review. This new method can be used in combination with experts' review to verify its sensitivity and specificity and to advance our understanding of the relationship between HFO and epilepsy. According to our data, the new method can be further developed as a fully automatic detector with high specificity and sensitivity.

The development of methods for analysis of a substantial amount of MEG/EEG data has become an important research area. For example, data mining has been developed for the analysis of HFOs in epilepsy patients (Blanco et al., 2011; Worrell et al., 2012). Blanco et al. (2011) reported a quantitative analysis of HFOs and their rates of occurrence in 9 patients with neocortical epilepsy and two control patients with no history of seizures (sampling rate: 32,556 Hz). Using the data mining approach, they found that a cluster of ripple frequency oscillations with a median spectral centroid of 137 Hz is increased in the seizure-onset zone more frequently than a cluster of fast ripple frequency oscillations (median spectral centroid = 305 Hz). Our results are consistent with their findings. The relative rate of ripple frequency oscillations is an interesting potential biomarker for the epileptic neocortex, but larger prospective studies correlating HFOs rates with seizure-onset zones, resected tissue and surgical outcomes are required to determine the true predictive value of this line of research (Montazeri et al., 2009; Blanco et al., 2011; Worrell et al., 2012). However, to our understanding, algorithmic requirements differ substantially for data mining and for topological (feature) data analysis. In particular, little is known about the locations of high frequency brain signals and their relationship to neurological disorders. In this regard, one of the unique features of accumulated source imaging is its ability to localize and visualize epileptic activity in both low- and high-frequency ranges for correlating locations of brain signals to neurological disorders.

Multi-modality imaging data challenge

Our data have also shown that functional MEG data can be seamlessly integrated into structural MRI data. In comparison to conventional source imaging, one important feature of accumulated source imaging is MPPL. MPPL analysis results in multi-values per voxel in 3D images. One parameter is dedicated to the frequency signature, which is important for visualizing HFOs. For example, HFOs may be a band-limited event (Crepon et al., 2010) or they can be a broadband event (Staba and Bragin, 2011; Worrell et al., 2012; Zijlmans et al., 2012b). By visualizing the frequency in the imaging data, we can better address many current questions in the study of HFOs (Engel et al., 2009; Staba and Bragin, 2011; Worrell et al., 2012; Zijlmans et al., 2012b). For example, if HFOs are band-limited, should there be specific spectral boundaries? In other words, should a HFO be defined as an isolated event in the time–frequency map, or could it contain a variety of frequencies within a range? Since spontaneous activity can occur in multi-frequency ranges, we consider the development of accumulated source imaging with MPPL to be important for multi-modality integration because the unique parameters from MEG is encoded in each location (voxel) and can be easily integrated into other modalities without losing any information.

Integration of accumulating and source localization into a systematic approach is a powerful neuroimaging data processing technique that could simplify multi-modality analyses. For example, epileptic foci defined by HFOs are not time-locked and can spontaneously occur at any time point or time window.

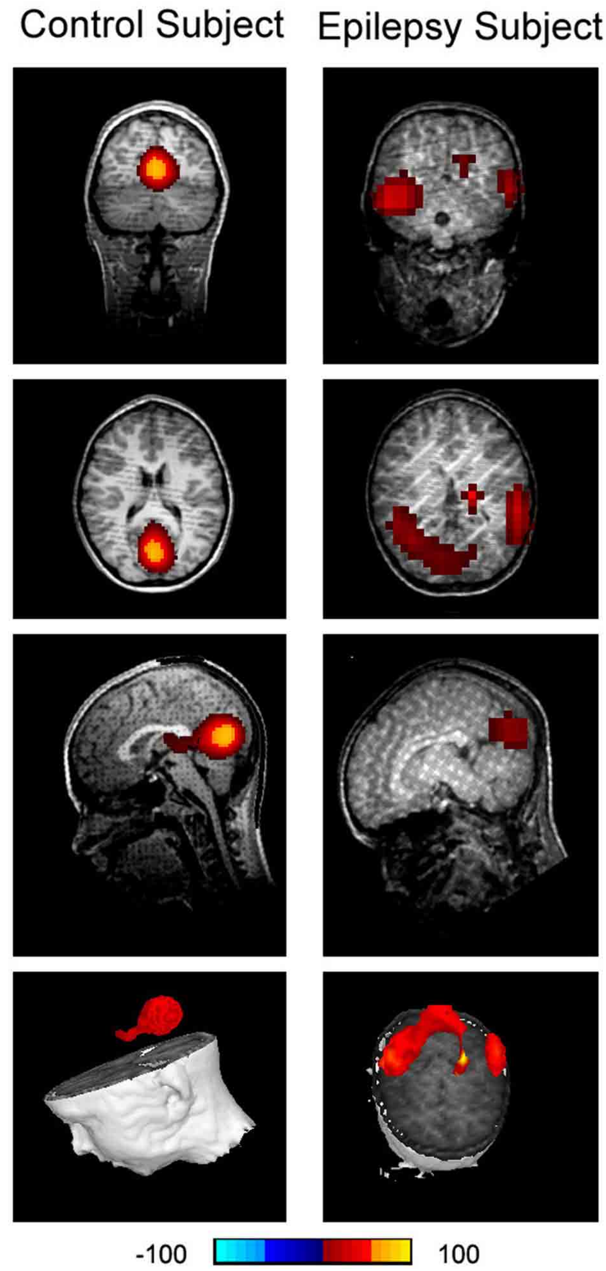


FIGURE 8 | Accumulated source imaging shows low frequency brain activity in 8–12 Hz (alpha) in an epilepsy subject ("Epilepsy Subject") and a healthy subject ("Control Subject"). Alpha activity is localized to the occipital cortex in the healthy subject. However, alpha activity is overshadowed by epileptic activity in the epilepsy subject. The epileptic activity is localized to the left and right parietal cortices in the epilepsy subject, which is concordant with clinical findings.

Without accumulation, thousands of MEG source images may need to be integrated into structural MRIs, which would be time-consuming. With accumulated source imaging, epileptic foci can be captured and summarized as a few images, which could be easily integrated into structural MRIs. As demonstrated in the Results section, accumulated source imaging is a potential powerful technique for multi-modality analysis of epileptic foci. Importantly, our software packages and libraries are based on C/C++. This methodology can be similarly implemented in more advanced computer systems such as cluster/GPU/FPGA or cloud/HPC. By using those advanced computer technologies, accumulated source imaging can be computed in a timely manner and routinely used in clinical practice in the future.

Clinical applicability challenge

Building on previous reports (Robinson et al., 2004; Kirsch et al., 2006) and our clinical observation, we developed the aforementioned method for detecting both low- and high-frequency brain signals. The proposed framework and architecture may also solve a few problems occurring in our clinical practice. First, this method provides an objective means of data analysis. The existing and currently practiced method for identifying epileptic spikes relies on visual inspection, which is subjective. Second, the proposed method provides meaningful quantitative source data, which are not available in conventional visual identification of epileptic spikes. Third, the new method can provide novel frequency descriptions about aberrant brain activity. In addition, the newly developed method semi-automatically quantifies MEG spectral power and source activity. Moreover, the new method has the capability of detecting and localizing high-frequency epileptic signals, a feat impossible to achieve with the conventional visual inspection of waveforms.

The results of spectral data showed that accumulated source imaging may play a key role in the differentiation of true HFOs from environmental noise in pre-operative workup for epilepsy surgery. It is well known that low-frequency signals may generate high-frequency harmonics. Since any harmonic of a high-frequency signal will localize to the corresponding low-frequency component, accumulated source imaging (which encodes both frequency and spatial information) can automatically reveal the main frequency by comparing the spectral power in the location of question. If HFOs were localized to a brain area which did not have low-frequency signals, the location would be an index for true HFOs. Since digital filtering may be used in the analysis of HFOs, the filter characteristics must be taken into account to avoid the detection of false oscillations (Benar et al., 2010). It has been noted that sharp transients with spectral content in HFO bands but without actual HFO in the raw data may be generated by filtering. According to our observation, such false oscillations are typically the result of the additive superposition of harmonics. They do not have a consistent spatial pattern in CxC and cannot be consistently localized to a location in the brain.

Accumulated source imaging may also play a key role in the differentiation of brain HFOs from artifacts in clinical practice. Raw MEG data contain a mixture of high-frequency brain signals and a variety of artifacts and noise. A major obstacle to HFO research is the unfortunate fact that various muscle activities typically result in prominent increases in gamma power (>25 Hz), and contaminate the recorded signal in the HFO spectrum. Myogenic activity interferes with the detection of HFO and represents a significant and often under-estimated challenge in clinical and basic research. For many years, intracranial EEG

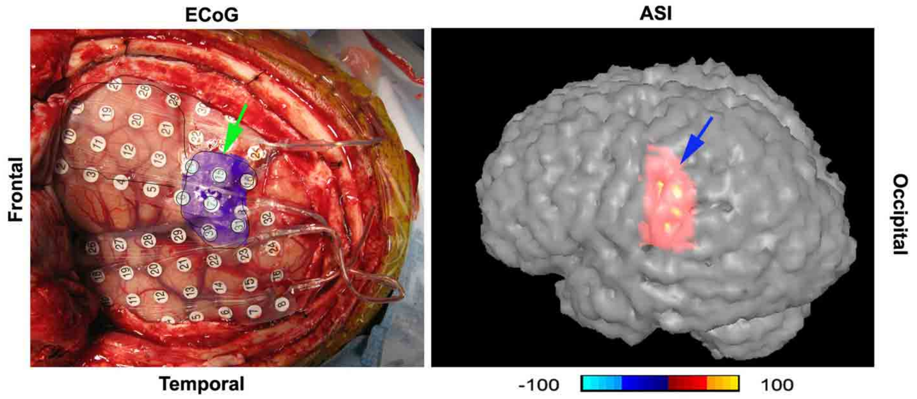


FIGURE 9 | A digital photo of intracranial recording ("ECoG") and accumulated source imaging ("ASI") show the concordance of the two methods. The two images are placed in the similar orientation. "Frontal" indicate the frontal cortex; "Temporal" indicates temporal lobe. The left green arrow points to the epileptic area invasively defined by intracranial recording; the right blue arrow points to the epileptic area noninvasively localized with high-frequency neuromagnetic signals. Noticeably, the areas are matched in gyrus level. The color bar shows the color coding of accumulated source imaging. The value of the source voxel is normalized T value (no unit).

Table 2 | The sensitivity and specificity of five MEG source localization methods.

		ASI	DM	BF	MN	MUSIC
1–70 Hz	Sensitivity	72	50	60	70	70
	Specificity	64	38	59	60	54
70–200 Hz	Sensitivity	90	30	40	50	40
	Specificity	76	42	53	62	68

recordings were assumed to be largely, if not completely, immune to eye movements and muscle artifacts. This assumption has recently been proven to be erroneous (Ball et al., 2009; Jerbi et al., 2009; Kovach et al., 2011). To solve these problems, we tried a different approach. Since high-frequency signals are typically obscured by low-frequency signals, time-frequency representations could be normalized according to the magnitude of each frequency bin across all MEG sensors to ensure that all frequency bins contributed equally to the source reconstruction. The time-frequency matrix allows for matrix operations to maximize the signal power at a peak frequency while simultaneously minimizing it at the neighboring surrounding frequency bins. CxC matrices based on the time-frequency data provide spatial patterns and gradients of magnetic fields for accurately determining epileptic foci for clinical purposes. In addition, the method was able to show multiple metrics of source analyses. It is important to be able to visualize different metrics of the source data because the frequency, strength, reliability/probability and kurtosis are important for us to correctly interpret the results. According to our pilot data, very strong high-frequency sources (>100 Hz) typically pinpointed to the epileptogenic zones and the removal of these zones would likely result in good surgical outcomes and ultimately seizure freedom (Xiang et al., 2009a). Thus, we postulate that these multiple metrics of source data will allow discrimination among pathological, benign or artifactural source signals in the future.

Although our newly developed method showed promising results for detecting both low- and high-frequency brain signals, several weaknesses and problems have been identified and need to be addressed in the future. Specifically, we used multiple local spheres in the computation of forward solution, which did not address the effect of the inferior conductive boundary of the skull that is not proximal to any MEG sensors. This leads to questions as to the accuracy of the forward model for "deep" sources, as may be encountered in temporal lobe epilepsy. A model based on the superior and lateral curvature of the head may mitigate this problem. The Sarvas forward solution, as applied to each sensor's sphere origin could only compute the field due to the tangential components of the dipole moment. Given that there were multiple sphere origins that might be in the vicinity of one another, the dipole orientation and therefore the weights might be "confused" by the rapid change (with location) of the tangential orientation. The number of sensors in node-beam lead field was experimentally determined by using from 3 to 275 sensors. Since three sensors have only two degrees of freedom left to attenuate unwanted interference or brain signal, at least five sensors are necessary to discriminate between brain source and interference (with 3rd gradient compensation). Of note, we continue to perform research into improving our methodology to overcome some of these limitations.

We also noted that source activities in a few subjects were close to the brain-stem. Those activities might be an artifact or localization problem or real sources. According to our data, the activity in the center of the brain is more than likely real for several reasons: (1) MEG data recorded without subjects did not show similar sources. Thus, it is unlikely that the sources are from system artifacts (e.g., hardware, software or localization algorithms); (2) the shape of the volumetric sources appears to mimic the structure of individual structural MRIs in subjects. If the deep sources are system artifacts or localization problems, they should not mimic the structure of individual MRI. (3) There are reports showing that MEG can detect and localize source in the deep brain areas. (4)

We were very careful to exclude artifact by excluding subjects with magnetic artifacts, recording MEG data with third-gradient noise cancellation and by visually inspecting the MEG data. However, MEG was not sensitive to sources in the brain areas and the source images of the deep source are more diffuse as compared to the surface sources. Therefore, further investigation and verification are necessary. We consider that closely following the guidance recommended by Gross and colleagues may further improve the quality of the data (Gross et al., 2013). In particular, by using multiple metrics of source analysis, we can incorporate new algorithms into data analysis by adding one source parameter. For example, Fatima and colleagues have developed a novel method to significantly improve the detection and localization of MEG sources by using independent component analysis (ICA) (Fatima et al., 2013), which can be incorporated into our source analysis pipeline to correct artifact and improve source localization. For resampling, one could also reduce the number of samples by high pass filtering the raw data and heterodyning it down to baseband, followed by decimation. The software and supplementary materials, which implemented the aforementioned algorithms, are freely available from the following website (http://sdrv.ms/PHenGK) for other researchers to test, reproduce, and improve the methods.

SUMMARY

In summary, the present study has demonstrated that accumulated source imaging is a new powerful technique for quantitatively and objectively analyzing MEG signals at source levels. By volumetrically scanning sources and accumulating source information, accumulated source imaging could handle very large datasets and extract meaningful spatial information about brain activity. Accumulated source imaging based on HFO detection may play a key role in differentiating brain activity from environmental noise and muscle artifacts. Though further verification is necessary, we believe that the next study should focus on using more advanced computer systems such as cluster/GPU/FPGA/cloud/HPC to significantly improve the performance of the proposed methods for clinical applications in the future.

AUTHOR CONTRIBUTIONS

Jing Xiang: study design and methodological development. Hisako Fujiwara, Nat Hemasilpin, Douglas F. Rose: data acquisition. Abraham Korman, Fawen Zhang, Jing Xiang: data analysis and manuscript preparation. Qian Luo, Rupesh Kotecha, Abraham Korman, Huan Luo: manuscript preparation.

ACKNOWLEDGMENTS

This study was partially supported by a Trustee Grant from Cincinnati Children's Hospital Medical Center and Grant Number R21NS072817 from the National Institute of Neurological Disorders and Stroke (NINDS), the National Institutes of Health. We thank Drs. Lu Tang, Hongxing Liu, Xiaoshan Wang and Jingping Shi for helping with data analyses and sharing MEG and clinical data. We are very grateful for the two anonymous reviewers for very important comments and suggestions for improving the manuscript and improving our methods.

REFERENCES

Agirre-Arrizubieta, Z., Huiskamp, G. J., Ferrier, C. H., Van Huffelen, A. C., and Leijten, F. S. (2009). Interictal magnetoencephalography and the irritative zone in the electrocorticogram. *Brain* 132, 3060–3071. doi: 10.1093/brain/awp137

Andrade-Valenca, L., Mari, F., Jacobs, J., Zijlmans, M., Olivier, A., Gotman, J., et al. (2012). Interictal high frequency oscillations (HFOs) in patients with focal epilepsy and normal MRI. *Clin. Neurophysiol.* 123, 100–105. doi: 10.1016/j.clinph.2011.06.004

Ball, T., Kern, M., Mutschler, I., Aertsen, A., and Schulze-Bonhage, A. (2009). Signal quality of simultaneously recorded invasive and non-invasive EEG. *Neuroimage* 46, 708–716. doi: 10.1016/j.neuroimage.2009.02.028

Benar, C. G., Chauviere, L., Bartolomei, F., and Wendling, F. (2010). Pitfalls of high-pass filtering for detecting epileptic oscillations: a technical note on "false" ripples. *Clin. Neurophysiol.* 121, 301–310. doi: 10.1016/j.clinph.2009.10.019

Benbadis, S. R., Lafrance, W. C. Jr., Papandonatos, G. D., Korabathina, K., Lin, K., Kraemer, H. C., et al. (2009). Interrater reliability of EEG-video monitoring. *Neurology* 73, 843–846. doi: 10.1212/WNL.0b013e3181b78425

Blanco, J. A., Stead, M., Krieger, A., Stacey, W., Maus, D., Marsh, E., et al. (2011). Data mining neocortical high-frequency oscillations in epilepsy and controls. *Brain* 134, 2948–2959. doi: 10.1093/brain/awr212

Brinkmann, B. H., Bower, M. R., Stengel, K. A., Worrell, G. A., and Stead, M. (2009). Large-scale electrophysiology: acquisition, compression, encryption, and storage of big data. *J. Neurosci. Methods* 180, 185–192. doi: 10.1016/j.jneumeth.2009.03.022

Chen, Y., Xiang, J., Kirtman, E. G., Wang, Y., Kotecha, R., and Liu, Y. (2010). Neuromagnetic biomarkers of visuocortical development in healthy children. *Clin. Neurophysiol.* 121, 1555–1562. doi: 10.1016/j.clinph.2010.03.029

Crepon, B., Navarro, V., Hasboun, D., Clemenceau, S., Martinerie, J., Baulac, M., et al. (2010). Mapping interictal oscillations greater than 200 Hz recorded with intracranial macroelectrodes in human epilepsy. *Brain* 133, 33–45. doi: 10.1093/brain/awp277

Dalal, S. S., Guggisberg, A. G., Edwards, E., Sekihara, K., Findlay, A. M., Canolty, R. T., et al. (2008). Five-dimensional neuroimaging: localization of the time-frequency dynamics of cortical activity. *Neuroimage* 40, 1686–1700. doi: 10.1016/j.neuroimage.2008.01.023

De Gooijer-Van De Groep, K. L., Leijten, F. S., Ferrier, C. H., and Huiskamp, G. J. (2013). Inverse modeling in magnetic source imaging: comparison of MUSIC, SAM(g2), and sLORETA to interictal intracranial EEG. *Hum. Brain Mapp.* 34, 2032–2044. doi: 10.1002/hbm.22049

De Munck, J. C., and Bijma, F. (2009). Three-way matrix analysis, the MUSIC algorithm and the coupled dipole model. *J. Neurosci. Methods* 183, 63–71. doi: 10.1016/j.jneumeth.2009.06.040

Dumpelmann, M., Jacobs, J., Kerber, K., and Schulze-Bonhage, A. (2012). Automatic 80–250 Hz "ripple" high frequency oscillation detection in invasive subdural grid and strip recordings in epilepsy by a radial basis function neural network. *Clin. Neurophysiol.* 123, 1721–1731. doi: 10.1016/j.clinph.2012.02.072

Engel, J. Jr., Bragin, A., Staba, R., and Mody, I. (2009). High-frequency oscillations: what is normal and what is not? *Epilepsia* 50, 598–604. doi: 10.1111/j.1528-1167.2008.01917.x

Fatima, Z., Quraan, M. A., Kovacevic, N., and McIntosh, A. R. (2013). ICA-based artifact correction improves spatial localization of adaptive spatial filters in MEG. *Neuroimage* 78, 284–294. doi: 10.1016/j.neuroimage.2013.04.033

Ghuman, A. S., Mcdaniel, J. R., and Martin, A. (2011). A wavelet-based method for measuring the oscillatory dynamics of resting-state functional connectivity in MEG. *Neuroimage* 56, 69–77. doi: 10.1016/j.neuroimage.2011.01.046

Gotman, J. (2010). High frequency oscillations: the new EEG frontier? *Epilepsia* 51(Suppl. 1), 63–65. doi: 10.1111/j.1528-1167.2009.02449.x

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., et al. (2014). MNE software for processing MEG and EEG data. *Neuroimage* 86, 446–460. doi: 10.1016/j.neuroimage.2013.10.027

Gross, J., Baillet, S., Barnes, G. R., Henson, R. N., Hillebrand, A., Jensen, O., et al. (2013). Good practice for conducting and reporting MEG research. *Neuroimage* 65, 349–363. doi: 10.1016/j.neuroimage.2012.10.001

Guggisberg, A. G., Dalal, S. S., Findlay, A. M., and Nagarajan, S. S. (2007). High-frequency oscillations in distributed neural networks reveal the dynamics of human decision making. *Front. Hum. Neurosci* 1:14. doi: 10.3389/neuro.09.014.2007

Gummadavelli, A., Wang, Y., Guo, X., Pardos, M., Chu, H., Liu, Y., et al. (2013). Spatiotemporal and frequency signatures of word recognition in the developing brain: a magnetoencephalographic study. *Brain Res.* 1498, 20–32. doi: 10.1016/j.brainres.2013.01.001

Haegelen, C., Perucca, P., Chatillon, C. E., Andrade-Valenca, L., Zelmann, R., Jacobs, J., et al. (2013). High-frequency oscillations, extent of surgical resection, and surgical outcome in drug-resistant focal epilepsy. *Epilepsia* 54, 848–857. doi: 10.1111/epi.12075

Hamalainen, M. S., and Sarvas, J. (1987). Feasibility of the homogeneous head model in the interpretation of neuromagnetic fields. *Phys. Med. Biol.* 32, 91–97. doi: 10.1088/0031-9155/32/1/014

Huang, M. X., Mosher, J. C., and Leahy, R. M. (1999). A sensor-weighted overlapping-sphere head model and exhaustive head model comparison for MEG. *Phys. Med. Biol.* 44, 423–440. doi: 10.1088/0031-9155/44/2/010

Huang, M. X., Shih, J. J., Lee, R. R., Harrington, D. L., Thoma, R. J., Weisend, M. P., et al. (2004). Commonalities and differences among vectorized beamformers in electromagnetic source imaging. *Brain Topogr.* 16, 139–158. doi: 10.1023/B:BRAT.0000019183.92439.51

Jacobs, J., Staba, R., Asano, E., Otsubo, H., Wu, J. Y., Zijlmans, M., et al. (2012). High-frequency oscillations (HFOs) in clinical epilepsy. *Prog. Neurobiol.* 98, 302–315. doi: 10.1016/j.pneurobio.2012.03.001

Jacobs, J., Zelmann, R., Jirsch, J., Chander, R., Dubeau, C. E., and Gotman, J. (2009). High frequency oscillations (80–500 Hz) in the preictal period in patients with focal seizures. *Epilepsia* 50, 1780–1792. doi: 10.1111/j.1528-1167.2009.02067.x

Jerbi, K., Freyermuth, S., Dalal, S., Kahane, P., Bertrand, O., Berthoz, A., and Lachaux, J. P. (2009). Saccade related gamma-band activity in intracerebral EEG: dissociating neural from ocular muscle activity. *Brain Topogr.* 22, 18–23. doi: 10.1007/s10548-009-0078-5

Jirsch, J. D., Urrestarazu, E., Levan, P., Olivier, A., Dubeau, F., and Gotman, J. (2006). High-frequency oscillations during human focal seizures. *Brain* 129, 1593–1608. doi: 10.1093/brain/awl085

Kirsch, H. E., Robinson, S. E., Mantle, M., and Nagarajan, S. (2006). Automated localization of magnetoencephalographic interictal spikes by adaptive spatial filtering. *Clin. Neurophysiol.* 117, 2264–2271. doi: 10.1016/j.clinph.2006.06.708

Kotecha, R., Xiang, J., Wang, Y., Huo, X., Hemasilpin, N., Fujiwara, H., et al. (2009). Time, frequency and volumetric differences of high-frequency neuromagnetic oscillation between left and right somatosensory cortices. *Int. J. Psychophysiol.* 72, 102–110. doi: 10.1016/j.ijpsycho.2008.10.009

Kovach, C. K., Tsuchiya, N., Kawasaki, H., Oya, H., Howard, M. A. 3rd., and Adolphs, R. (2011). Manifestation of ocular-muscle EMG contamination in human intracranial recordings. *Neuroimage* 54, 213–233. doi: 10.1016/j.neuroimage.2010.08.002

Le Van Quyen, M., Staba, R., Bragin, A., Dickson, C., Valderrama, M., Fried, I., et al. (2010). Large-scale microelectrode recordings of high-frequency gamma oscillations in human cortex during sleep. *J. Neurosci.* 30, 7770–7782. doi: 10.1523/JNEUROSCI.5049-09.2010

Levesque, M., Bortel, A., Gotman, J., and Avoli, M. (2011). High-frequency (80–500 Hz) oscillations and epileptogenesis in temporal lobe epilepsy. *Neurobiol. Dis.* 42, 231–241. doi: 10.1016/j.nbd.2011.01.007

Matsumoto, A., Brinkmann, B. H., Matthew Stead, S., Matsumoto, J., Kucewicz, M. T., Marsh, W. R., et al. (2013). Pathological and physiological high-frequency oscillations in focal human epilepsy. *J. Neurophysiol.* 110, 1958–1964. doi: 10.1152/jn.00341.2013

Montazeri, N., Shamsollahi, M. B., and Hajipour, S. (2009). MEG based classification of wrist movement. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2009, 986–989. doi: 10.1109/IEMBS.2009.5334472

Mosher, J. C., and Leahy, R. M. (1998). Recursive MUSIC: a framework for EEG and MEG source localization. *IEEE Trans. Biomed. Eng.* 45, 1342–1354. doi: 10.1109/10.725331

Mosher, J. C., Leahy, R. M., and Lewis, P. S. (1999). EEG and MEG: forward solutions for inverse methods. *IEEE Trans. Biomed. Eng.* 46, 245–259. doi: 10.1109/10.748978

Ou, W., Hamalainen, M. S., and Golland, P. (2009). A distributed spatio-temporal EEG/MEG inverse solver. *Neuroimage* 44, 932–946. doi: 10.1016/j.neuroimage.2008.05.063

Pail, M., Halamek, J., Daniel, P., Kuba, R., Tyrlikova, I., Chrastina, J., et al. (2013). Intracerebrally recorded high frequency oscillations: simple visual assessment versus automated detection. *Clin. Neurophysiol.* 124, 1935–1942. doi: 10.1016/j.clinph.2013.03.032

Papadelis, C., Poghosyan, V., Fenwick, P. B., and Ioannides, A. A. (2009). MEG's ability to localise accurately weak transient neural sources. *Clin. Neurophysiol.* 120, 1958–1970. doi: 10.1016/j.clinph.2009.08.018

Pulvermuller, F., Birbaumer, N., Lutzenberger, W., and Mohr, B. (1997). High-frequency brain activity: its possible role in attention, perception and language processing. *Prog. Neurobiol.* 52, 427–445. doi: 10.1016/S0301-0082(97)00023-3

Rau, R., Raschka, C., and Koch, H. J. (2002). Uniform decrease of alpha-global field power induced by intermittent photic stimulation of healthy subjects. *Braz. J. Med. Biol. Res.* 35, 605–611. doi: 10.1590/S0100-879X2002000500014

Restuccia, D., Del Piero, I., Martucci, L., and Zanini, S. (2011). High-frequency oscillations after median-nerve stimulation do not undergo habituation: a new insight on their functional meaning? *Clin. Neurophysiol.* 122, 148–152. doi: 10.1016/j.clinph.2010.06.008

Robinson, S. E. (2004). Localization of event-related activity by SAM(erf). *Neurol. Clin. Neurophysiol.* 2004, 109.

Robinson, S. E., Nagarajan, S. S., Mantle, M., Gibbons, V., and Kirsch, H. (2004). Localization of interictal spikes using SAM(g2) and dipole fit. *Neurol. Clin. Neurophysiol.* 2004, 74.

Sarvas, J. (1987). Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. *Phys. Med. Biol.* 32, 11–22. doi: 10.1088/0031-9155/32/1/004

Srejic, L. R., Valiante, T. A., Aarts, M. M., and Hutchison, W. D. (2013). High-frequency cortical activity associated with postischemic epileptiform discharges in an *in vivo* rat focal stroke model. *J. Neurosurg.* 118, 1098–1106. doi: 10.3171/2013.1.JNS121059

Staba, R. J., and Bragin, A. (2011). High-frequency oscillations and other electrophysiological biomarkers of epilepsy: underlying mechanisms. *Biomark. Med.* 5, 545–556. doi: 10.2217/bmm.11.72

Stacey, W. C., Kellis, S., Greger, B., Butson, C. R., Patel, P. R., Assaf, T., et al. (2013). Potential for unreliable interpretation of EEG recorded with microelectrodes. *Epilepsia* 54, 1391–1401. doi: 10.1111/epi.12202

Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., and Leahy, R. M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Comput. Intell. Neurosci.* 2011:879716. doi: 10.1155/2011/879716

Tort, A. B., Scheffer-Teixeira, R., Souza, B. C., Draguhn, A., and Brankack, J. (2013). Theta-associated high-frequency oscillations (110–160 Hz) in the hippocampus and neocortex. *Prog. Neurobiol.* 100, 1–14. doi: 10.1016/j.pneurobio.2012.09.002

Uhlhaas, P. J., Pipa, G., Neuenschwander, S., Wibral, M., and Singer, W. (2011). A new look at gamma? High- (>60 Hz) gamma-band activity in cortical networks: function, mechanisms and impairment. *Prog. Biophys. Mol. Biol.* 105, 14–28. doi: 10.1016/j.pbiomolbio.2010.10.004

Uhlhaas, P. J., and Singer, W. (2013). High-frequency oscillations and the neurobiology of schizophrenia. *Dialogues Clin. Neurosci.* 15, 301–313.

Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T. E., Bucholz, R., et al. (2012). The Human Connectome Project: a data acquisition perspective. *Neuroimage* 62, 2222–2231. doi: 10.1016/j.neuroimage.2012.02.018

Vrba, J., and Robinson, S. E. (2001). Signal processing in magnetoencephalography. *Methods* 25, 249–271. doi: 10.1006/meth.2001.1238

Weiss, S. A., Banks, G. P., McKhann, G. M. Jr., Goodman, R. R., Emerson, R. G., Trevelyan, A. J., et al. (2013). Ictal high frequency oscillations distinguish two types of seizure territories in humans. *Brain* 136, 3796–3808. doi: 10.1093/brain/awt276

Worrell, G. A., Jerbi, K., Kobayashi, K., Lina, J. M., Zelmann, R., and Le Van Quyen, M. (2012). Recording and analysis techniques for high-frequency oscillations. *Prog. Neurobiol.* 98, 265–278. doi: 10.1016/j.pneurobio.2012.02.006

Xiang, J., Degrauw, X., Korostenskaja, M., Korman, A. M., O'Brien, H. L., Kabbouche, M. A., et al. (2013). Altered cortical activation in adolescents with acute migraine: a magnetoencephalography study. *J. Pain* 14, 1553–1563. doi: 10.1016/j.jpain.2013.04.009

Xiang, J., Holowka, S., Qiao, H., Sun, B., Xiao, Z., Jiang, Y., et al. (2004). Automatic localization of epileptic zones using magnetoencephalography. *Neurol. Clin. Neurophysiol.* 2004, 98.

Xiang, J., Liu, Y., Wang, Y., Kirtman, E. G., Kotecha, R., Chen, Y., et al. (2009a). Frequency and spatial characteristics of high-frequency neuromagnetic signals in childhood epilepsy. *Epileptic Disord.* 11, 113–125. doi: 10.1684/epd.2009.0253

Xiang, J., Liu, Y., Wang, Y., Kotecha, R., Kirtman, E. G., Chen, Y., et al. (2009b). Neuromagnetic correlates of developmental changes in endogenous high-frequency brain oscillations in children: a wavelet-based beamformer study. *Brain Res.* 1274, 28–39. doi: 10.1016/j.brainres.2009.03.068

Xiang, J., Wang, Y., Chen, Y., Liu, Y., Kotecha, R., Huo, X., et al. (2010). Noninvasive localization of epileptogenic zones with ictal high-frequency neuromagnetic signals. *J. Neurosurg. Pediatr.* 5, 113–122. doi: 10.3171/2009.8.PEDS09345

Xiang, J., and Xiao, Z. (2009). Spatiotemporal and frequency signatures of noun and verb processing: a wavelet-based beamformer study. *J. Clin. Exp. Neuropsychol.* 31, 648–657. doi: 10.1080/13803390802448651

Zafeiriou, D. I., and Vargiami, E. (2012). Noninvasive ultra high-frequency (1kHz) oscillations' recording: high-fidelity over somatosensory cortex. *Clin. Neurophysiol.* 123, 2323–2324. doi: 10.1016/j.clinph.2012.05.010

Zijlmans, M., Huiskamp, G. M., Cremer, O. L., Ferrier, C. H., Van Huffelen, A. C., and Leijten, F. S. (2012a). Epileptic high-frequency oscillations in intraoperative electrocorticography: the effect of propofol. *Epilepsia* 53, 1799–1809. doi: 10.1111/j.1528-1167.2012.03650.x

Zijlmans, M., Jiruska, P., Zelmann, R., Leijten, F. S., Jefferys, J. G., and Gotman, J. (2012b). High-frequency oscillations as a new biomarker in epilepsy. *Ann. Neurol.* 71, 169–178. doi: 10.1002/ana.22548

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 23 December 2013; accepted: 02 May 2014; published online: 21 May 2014.
Citation: Xiang J, Luo Q, Kotecha R, Korman A, Zhang F, Luo H, Fujiwara H, Hemasilpin N and Rose DF (2014) Accumulated source imaging of brain activity with both low and high-frequency neuromagnetic signals. Front. Neuroinform. ***8****:57. doi: 10.3389/fninf.2014.00057*
This article was submitted to the journal Frontiers in Neuroinformatics.

METHODS ARTICLE
published: 31 January 2014
doi: 10.3389/fninf.2014.00002

A versatile software package for inter-subject correlation based analyses of fMRI

Jukka-Pekka Kauppi[1,2]†, *Juha Pajula*[3]† *and Jussi Tohka*[3]*

[1] Department of Computer Science and HIIT, University of Helsinki, Helsinki, Finland
[2] Brain Research Unit, O.V. Lounasmaa Laboratory, School of Science, Aalto University, Espoo, Finland
[3] Department of Signal Processing, Tampere University of Technology, Tampere, Finland

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Ralf Schmaelzle, University of Konstanz, Germany
Fani Deligianni, University College London, UK

***Correspondence:**
Jussi Tohka, Department of Signal Processing, Tampere University of Technology, PO Box 553, 33101 Tampere, Finland
e-mail: jussi.tohka@tut.fi

† *These authors have contributed equally to this work.*

In the inter-subject correlation (ISC) based analysis of the functional magnetic resonance imaging (fMRI) data, the extent of shared processing across subjects during the experiment is determined by calculating correlation coefficients between the fMRI time series of the subjects in the corresponding brain locations. This implies that ISC can be used to analyze fMRI data without explicitly modeling the stimulus and thus ISC is a potential method to analyze fMRI data acquired under complex naturalistic stimuli. Despite of the suitability of ISC based approach to analyze complex fMRI data, no generic software tools have been made available for this purpose, limiting a widespread use of ISC based analysis techniques among neuroimaging community. In this paper, we present a graphical user interface (GUI) based software package, ISC Toolbox, implemented in Matlab for computing various ISC based analyses. Many advanced computations such as comparison of ISCs between different stimuli, time window ISC, and inter-subject phase synchronization are supported by the toolbox. The analyses are coupled with re-sampling based statistical inference. The ISC based analyses are data and computation intensive and the ISC toolbox is equipped with mechanisms to execute the parallel computations in a cluster environment automatically and with an automatic detection of the cluster environment in use. Currently, SGE-based (Oracle Grid Engine, Son of a Grid Engine, or Open Grid Scheduler) and Slurm environments are supported. In this paper, we present a detailed account on the methods behind the ISC Toolbox, the implementation of the toolbox and demonstrate the possible use of the toolbox by summarizing selected example applications. We also report the computation time experiments both using a single desktop computer and two grid environments demonstrating that parallelization effectively reduces the computing time. The ISC Toolbox is available in https://code.google.com/p/isc-toolbox/

Keywords: functional magnetic resonance imaging, naturalistic stimulus, re-sampling test, Matlab, grid-computing, GUI

1. INTRODUCTION

Most neuroimaging studies, such as those based on functional magnetic resonance imaging (fMRI), have so far utilized relatively simple static stimuli to analyze brain functions (Spiers and Maguire, 2007). However, the human brain has evolved to function in a tremendously stimulating world and the investigation of complex brain functions, including socio-emotional or comprehension-related processes, is limited when using highly controlled/simplistic experimental setups, because these functions are only triggered under highly complex stimuli. There is an increasing interest in studying the human brain function with dynamic, continuous stimuli that are designed to be closer to normal everyday life than in conventional, strictly controlled research paradigms. The used stimuli can be, for example, a movie. This kind of fMRI data cannot be straight-forwardly analyzed based on a general linear model (GLM), because a GLM requires a reference time course of the task that is impossible to obtain for a multi-dimensional stimulus such as a movie, unless focusing the data-analysis on a specific feature of the stimuli. For this reason, new data-driven methodologies are needed. The use of novel experimental setups involving rich stimuli and data-driven analysis methods which are particularly designed to study complex brain functions opens up entire new fields for neuroscience research.

Inter-subject correlation (ISC) based analysis, originally introduced by Hasson et al. (2004), is a conceptually simple approach to analyze fMRI data acquired under naturalistic stimuli. In the ISC based analysis, the extent of shared processing across subjects during the experiment is determined by calculating correlation coefficient between the fMRI time series of the subjects in the corresponding brain locations. This way, ISC based analyses effectively avoid the modeling of the stimuli.

ISC based analyses have been previously applied to analyze fMRI data collected during complex stimuli or tasks, including movies (Hasson et al., 2004; Jääskeläinen et al., 2008; Kauppi et al., 2010b; Nummenmaa et al., 2012), TV news reports (Schmälzle et al., 2013), auditory and audiovisual narratives (Wilson et al., 2008), pieces of music (Abrams et al., 2013) and aesthetic performances (Jola et al., 2013). ISC based analysis has also been used for feature selection as a part of multivariate pattern analysis of data collected during a movie experiment (Kauppi et al., 2011). There can be different motivations to apply ISC based analysis for fMRI data. One can address specific neuroscientific research questions (some examples are provided in section 3) or simply try to make sense of highly complex fMRI data to generate new hypotheses. Whatever the motivation, it is important to keep in mind that the ISC is primarily a measure of *shared* hemodynamic activity across subjects and not a measure of hemodynamic activity *per se*. However, as shown by Pajula et al. (2012), when equipped with proper nonparametric statistical procedures (Kauppi et al., 2010b), ISC based methods can be used for detecting traditional fMRI activations without requiring specific, *a-priori* stimulus time course models.

Despite of the suitability of the ISC based approach to analyze complex fMRI data, no generic software tools have been made available for this purpose, limiting a widespread use of ISC based analysis techniques among neuroimaging community. Reliable and sophisticated ISC based analysis requires management of several nontrivial methodological, computational, and visualization related issues (such as heavy computational and memory load of the analysis, the choice of a proper ISC measure, handling nonstandard statistical significance testing, and the visualization of multidimensional time-varying ISC maps). Hence, it is obvious that a toolbox solving these issues would be highly beneficial and can substantially simplify the use of the ISC based analysis among neuroscientists, consecutively advancing our understanding of complex human brain functions.

We have previously introduced a framework for the basic ISC based analysis (Kauppi et al., 2010b) and started building an open source, graphical user interface (GUI) based Matlab toolbox, termed the ISC toolbox, for a generic, ISC based analysis of fMRI. A set of visualization tools—particularly designed for the ISC analyses—are integrated to the GUI. In this paper, we describe the methods behind of the ISC toolbox that implements, in addition to the basic ISC analysis, many advanced ISC based computations such as phase ISC, time-windowed ISC, and comparison of ISCs between different stimuli. We will describe the analysis methods, explain the rationales behind them and demonstrate their potential use by reviewing selected example application studies.

As the ISC based analyses are data and computation intensive, the ISC toolbox is equipped with mechanisms to execute the parallel computations in a cluster environment automatically and with an automatic detection of the cluster environment in use. Currently, SGE-based environments [Unity Grid Engine (Univa Corporation, 2013), Son of a Grid Engine (Love, 2013), or Open Grid Scheduler (Scalable Logic, 2013)] and Slurm environment (Yoo et al., 2003) are supported. As there are ISC method-specific challenges in the parallelization, we will describe the automatic parallelization mechanisms in the paper. The ISC toolbox (the current version is 2.0) is available in https://code.google.com/p/isc-toolbox/

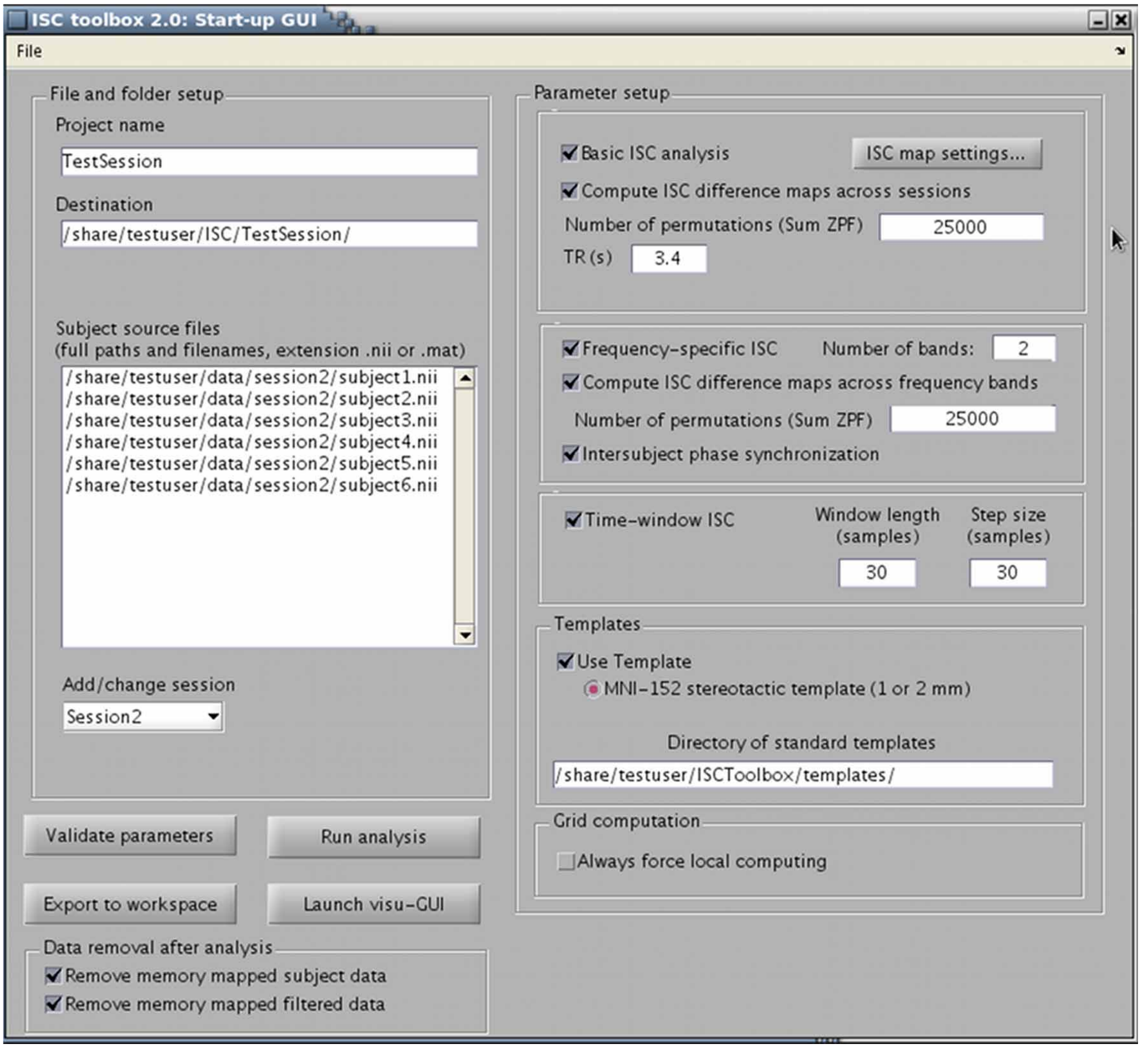
The organization of the paper is as follows. In section 2, after providing an overview of the toolbox, we will detail the ISC methods (section 2.2), describe the implementation of the toolbox (section 2.3), and briefly describe a set of visualization tools, customized to the ISC analyses (section 2.4). In section 3, we demonstrate the use of ISC-based analyses by reviewing selected studies. In section 4, as we consider cluster computing features of the toolbox important, we present the computation time experiments demonstrating the added value of parallel computing. Section 5 discusses current limitations and future directions of the toolbox and section 6 concludes the paper.

2. MATERIALS AND METHODS

2.1. OVERVIEW AND USAGE OF ISC TOOLBOX

The ISC toolbox is designed for generic ISC based analysis of fMRI data. No information about the stimulus is required to carry out the analysis, making the toolbox suitable to analyze nearly any kind of fMRI data. Naturally, data from at least two subjects are needed for the analysis because the analysis procedure is based on voxel-wise correlations of fMRI time-series across subjects. A normal desktop computer equipped with the Matlab is sufficient to carry out the basic ISC analysis in many situations. However, in certain situations it is recommended to utilize a computer cluster to carry out the analysis. For instance, the use of cluster can be meaningful if the number of subjects is high (tens of subjects), advanced ISC analyses need to be computed, or reliable re-sampling based nonparametric statistical inference is needed to construct ISC maps. The toolbox can efficiently and automatically utilize cluster environment, allowing easy and fast ISC based analysis.

The toolbox consists of three parts: (1) a startup GUI for setting-up parameters for the analysis, (2) a main program that computes ISC maps based on selected parameters, and (3) a GUI-based visualization tool for the exploration of the findings. The GUIs are designed to make the analysis easier but a whole analysis pipeline can also be carried out from Matlab's command line. The main window of the startup GUI is shown in **Figure 1** to demonstrate the main features of the ISC toolbox. Using the startup GUI, a user can easily select the appropriate analyses and their parameters. In the left side of the panel, a user chooses a descriptive project name and the destination folder of the analysis. For a large textbox ("Subject source files"), a user adds the names of the files containing fMRI time-series of the subjects used in the analysis. The toolbox assumes that fMRI signals have been preprocessed and preferably registered to a standard template. Preprocessing and registration algorithms are not implemented in the ISC toolbox because well developed free software packages exist for these purposes. Preprocessed and registered fMRI data sets of the subjects should be given either in nifti- or mat-format as 4-dimensional (a 3-dimensional position coordinate and time) matrices. If several acquisitions are available for each subject or acquisitions for more than one group are available, a user can analyze them all by adding more sessions to the project. The left side of the panel also contains buttons for parameter validation and for launching the main program which computes ISC maps once the parameters have been successfully validated. After running the main program, the visualization GUI to analyze results can be launched from the separate button. There is also an option to export parameters to Matlab's workspace (using the button "Export to workspace"). Automatic postprocessing



ISC toolbox 2.0: Start-up GUI

File

File and folder setup

Project name: TestSession

Destination: /share/testuser/ISC/TestSession/

Subject source files (full paths and filenames, extension .nii or .mat)

```
/share/testuser/data/session2/subject1.nii
/share/testuser/data/session2/subject2.nii
/share/testuser/data/session2/subject3.nii
/share/testuser/data/session2/subject4.nii
/share/testuser/data/session2/subject5.nii
/share/testuser/data/session2/subject6.nii
```

Add/change session: Session2

Parameter setup

☑ Basic ISC analysis

☑ Compute ISC difference maps across sessions

Number of permutations (Sum ZPF): 25000

TR (s): 3.4

☑ Frequency-specific ISC Number of bands: 2

☑ Compute ISC difference maps across frequency bands

Number of permutations (Sum ZPF): 25000

☑ Intersubject phase synchronization

☑ Time-window ISC Window length (samples): 30 Step size (samples): 30

Templates

☑ Use Template

● MNI-152 stereotactic template (1 or 2 mm)

Directory of standard templates: /share/testuser/ISCToolbox/templates/

Grid computation

□ Always force local computing

Data removal after analysis

☑ Remove memory mapped subject data

☑ Remove memory mapped filtered data

FIGURE 1 | ISC Toolbox's startup GUI where user can define the parameters for analysis, test them, run the ISC based computations and launch a separate GUI for visualization of the results.

operations can also be used to remove portions of data generated during the analysis to free disk space.

Different ISC analysis options are selected from the right side of the panel. A *basic ISC analysis* includes the generation of the ISC maps including thresholding of the maps based on a nonparametric statistical test. Details of this analysis can be specified from a separate panel under a button "ISC map settings." If more than one session is added to the project, it is possible to compute *ISC difference maps* to investigate whether ISCs in some of the sessions (conditions) are higher than in the others. *Frequency-specific ISC* decomposes fMRI time-series of the subjects to frequency sub-bands and computes and thresholds ISC maps for each sub-band. *Time-window ISC* computes ISC maps for several consecutive time-frames. *Inter-subject phase synchronization* combines the localization of inter-subject similarities in space, time, and frequency. These analyses are explained in section 2.2.

An arbitrary volume size can be used to compute ISC maps as long as the volume is same across subjects. However, the GUI built for the visualization of the results assumes that all fMRI data sets have been registered to a common MNI152 template. The toolbox also assumes that Harvard-Oxford cortical and sub-cortical brain atlases are available to compute and visualize inter-subject similarities for selected brain regions. Hence, to allow convenient analysis of the results, it is highly recommended to register the data to the MNI template prior to ISC analysis as well as to have the Harvard-Oxford brain atlases available. The anatomical template, atlases, and the brain mask for limiting ISC computations only for the voxels within the brain are freely provided with the FSL software package. The directory including the corresponding nifti-files should be provided in the startup GUI (subpanel "Templates"). The use of a computational cluster can be disabled under the panel "Grid computation" if needed.

After all parameters have been set and validated, they are automatically saved under the project directory in a single structure array called "Params" (the parameters can also be saved or the existing parameters can be loaded by a user from the file-menu in the upper left corner). The main program performs all ISC based computations defined in this parameter structure array. The program saves intermediate and final results of the computations to the project folders. The visualization GUI allows flexible analysis of ISC maps over an anatomical template together with the brain atlases. It also allows exporting interesting data to Matlab's Workspace for customized analysis.

2.2. ISC METHODS

2.2.1. Generation and visualization of the ISC maps

A correlation coefficient [1] is a natural measure of similarity between fMRI time-courses of two subjects. ISC toolbox allows an analysis of the similarities in the time-courses across multiple subjects. We compute the mean of the voxel-wise correlation coefficients across all possible subject pairs as (Kauppi et al., 2010b):

$$\bar{r} = \frac{1}{N(N-1)/2}\sum_{i=1}^{N}\sum_{j=2,j>i}^{N-1} r_{ij}, \tag{1}$$

where $\bar{r}$ denotes a group-level ISC in a given voxel (a voxel index is omitted for clarity), N is the total number of subjects, and r_{ij} is the correlation coefficient between fMRI time-courses of subjects i and j. Note that because $r_{ii} = 1$ and $r_{ij} = r_{ji}$, it is sufficient to compute correlation coefficients across $N(N-1)/2$ subject pairs (instead of N^2 pairs). However, because the number of subject pairs increases approximately quadratically with N and Equation (1) is computed for every voxel within the brain, it may be necessary to compute extremely high number of correlation coefficients (in the order of 10^8) even for the most basic ISC analysis, rendering the analysis procedure computationally demanding.

We briefly explain our preference to $\bar{r}$ as the test-statistic, particularly over a related one used by Lerner et al. (2011). The main reason is that the test statistic $\bar{r}$ can be seen as an estimator of the true (but unknown) population ISC ρ under the model that $\rho_{ij} = \rho + \epsilon_{ij}$, where ρ_{ij} is the true correlation between subjects i and j and ϵ_{ij}, with zero-expectation, models the between subject-pair variation. More specifically, if r_{ij} approaches ρ_{ij} and ρ_{ij} approaches ρ, then $\bar{r}$ approaches ρ. Lerner et al. (2011) computed the average correlation of the subject time course and average time course of remaining subjects. This is closely related to $\bar{r}$ statistic[2] and neither one seems to be quantitatively better than the other. However, the statistic in Lerner et al. (2011) cannot be straight-forwardly interpreted as an estimator of the population ISC in an above sense, which results in our preference of $\bar{r}$.

2.2.2. Nonparametric re-sampling test

The correlation coefficients r_{ij} in Equation (1) are not independent because each subject is present in more than one subject pair (e.g., r_{ij} and r_{kj} are overlapping because they both depend on the same time-series measured from subject j). Also, it is well known that BOLD-fMRI signals are temporally correlated. Therefore, the standard tests for assessing the significance of $\bar{r}$ are not valid. We use a fully nonparametric re-sampling based method to evaluate the significance of $\bar{r}$ (Kauppi et al., 2010b). In this method, we perform a test against a null hypothesis that $\bar{r}$ statistic is the same as for data with no specific time-structure. To compute a "null" re-sampling distribution, we circularly shift each subjects time-series by a random amount so that they are no longer aligned in time across the subjects, and then calculate $\bar{r}$ statistic. This way we can account for temporal autocorrelations present in the fMRI data. In practice, calculation of all the possible time shift combinations is computationally prohibitive and the distribution is approximated with finite number of realizations, randomizing the experiment across voxels and time-points, by default 100 million realizations are generated. To obtain critical thresholds for significant ISCs, we first compute p-values of the true realizations for each voxel based on the null distribution and then correct the values using the false discovery rate (FDR) based multiple comparisons correction (Benjamini and Hochberg, 1995). Using our visualization tool, it is possible to investigate thresholded ISC maps over an anatomical template with different critical thresholds.

2.2.3. Parametric t-test

The ISC toolbox contains an option to threshold group-level ISC maps also based on a simple parametric test proposed by Wilson et al. (2008). For this test, correlation coefficients are first transformed to z-scores using a Fisher's z transformation:

$$z_{ij} = \frac{1}{2}\log\left(\frac{1+r_{ij}}{1-r_{ij}}\right). \tag{2}$$

Then, a one-sample t-test with $N(N-1)/2 - 1$ degrees of freedom is performed under a null hypothesis that the ISC is zero. Note that the independence assumption of the observations made by the test is violated in practice.

[1] By correlation coefficient, we refer to a standard Pearsons correlation coefficient.

[2] Let $\mathbf{s}_i$ denote the time course of the subject i that is de-meaned and normalized to unit length so that $||\mathbf{s}_i|| = 1$ (the conclusion of this analysis does not depend on the normalization to the unit length but the analysis is simplified by that assumption). Now, r_{ij} can be written as an inner-product $r_{ij} = \mathbf{s}_i^T\mathbf{s}_j$. Define a test statistic similarly to Lerner et al. (2011)

$$\bar{l} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i^T\left(\frac{1}{Z_{-i}(N-1)}\sum_{j=1,\,j\neq i}^{N}\mathbf{s}_j\right),$$

where $Z_{-i} = ||(1/(N-1))\sum_{j=1,\,j\neq i}^{N}\mathbf{s}_j||$. A straight-forward computation yields

$$\bar{l} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i^T\left(\frac{1}{(N-1)Z_{-i}}\sum_{j=1,\,j\neq i}^{N}\mathbf{s}_j\right) = \frac{1}{N}\frac{1}{N-1}\sum_{i=1}^{N}\sum_{j\neq i}\frac{\mathbf{s}_i^T\mathbf{s}_j}{Z_{-i}}$$

$$= \frac{1}{N^2-N}\sum_{i=1}^{N}\sum_{j\neq i}\frac{r_{ij}}{Z_{-i}} = \frac{1}{\frac{N^2-N}{2}}\sum_{i=1}^{N}\sum_{j=2,\,j>i}^{N}\frac{r_{ij}}{Z_{-i}},$$

since $r_{ij} = r_{ji}$. It can be seen that this is a weighted version of $\bar{r}$, where the weights are proportional to the standard deviations of the average time courses.

2.2.4. Generation and visualization of the ISC difference maps

With the ISC toolbox, it is also possible to generate and visualize ISC *difference maps* to investigate if there are significant differences in the ISCs between two conditions. For instance, in studies where same subjects are scanned twice under different stimuli, it can be highly interesting to analyze whether or not ISC was stronger in one of the conditions. We use a modified Pearson-Filon statistic based on Fisher's z-transformation (ZPF; Raghunathan et al., 1996) for this purpose, which is a recommended statistic for testing if two nonoverlapping but dependent correlation coefficients are different (Krishnamoorthy and Xia, 2007). Consider four time-series $\mathbf{t}_i^a$, $\mathbf{t}_j^a$, $\mathbf{t}_i^b$ and $\mathbf{t}_j^b$ measured from two subjects i and j in two conditions a and b. The corresponding correlation coefficients r_{ij}^a and r_{ij}^b are nonoverlapping, because they have been computed using different time-series. However, stimuli used in two conditions a and b may not be independent, making a dependency assumption plausible. We extend the pairwise ZPF statistic for group-level analysis by combining the pairwise statistic from all subject pairs, and design a fully nonparametric test to assess the significance of the resulting group-level statistic (Reason et al., under review). Our final "sum ZPF" statistic is given by:

$$\mathrm{ZPF}_{\Sigma ij}^{ab} = \sum_{i=1}^{N} \sum_{j=2,\, j>i}^{N-1} \frac{(z_{ij}^a - z_{ij}^b)\sqrt{(T-3)/2}}{\sqrt{1-\mathrm{cov}(r_{ij}^a, r_{ij}^b) \Big/ \Big[\Big(1-(r_{ij}^a)^2)(1-(r_{ij}^b)^2\Big)\Big]}}, \quad (3)$$

where z_{ij}^a, z_{ij}^b are the Fisher's z transforms [see Equation (2)] of the correlation coefficients r_{ij}^a, r_{ij}^b, respectively, T is the length of a time-course and $\mathrm{cov}(r_{ij}^a,\ r_{ij}^b)$ is a large scale covariance (Raghunathan et al., 1996). The test is performed under the null hypothesis that each ZPF value is drawn from a distribution with zero mean, which occurs when there is no difference in ISC between the conditions. The approximate permutation distribution is generated by randomly flipping the sign of pairwise ZPF statistics before calculating Equation (3) using a subsample of all possible random labelings. Maximal and minimal statistics over the entire image corresponding to each labeling are saved to account for multiple comparisons by controlling family-wise error rate (FWER; Nichols and Holmes, 2002). Due to the symmetry of the distribution, thresholds for both directions are obtained with this procedure. The default number of random permutations over the whole image is 25,000.

Note that we cannot readily confirm the full exchangeability under the null hypothesis for the permutation test since: (1) fMRI time series are autocorrelated and (2) the subject pairs are not independent. Assuming temporal independence and normality, the ZPF-statistic can be shown to be distributed according to the standard normal distribution under the null hypothesis of no correlation difference (Raghunathan et al., 1996), which is enough to ensure the correctness of the test (Good, 2005). However, it is unclear to what extent this distributional result holds for the ISC analysis. We performed here a simple Monte Carlo simulation that verified that the ZPF statistics are normally distributed with a constant variance, not dependent on the (true) values of $r_{ij}^a = r_{ij}^b$, thus partially verifying the permutation test. The experiment and its results are summarized in **Figure 2**.

2.2.5. Frequency-specific ISC analysis

The ISC toolbox contains an option to analyze ISCs in distinct frequency sub-bands. The approach is well-motivated because real-world events and stimuli unfold over multiple time-scales (Kauppi et al., 2010b). For instance, features of visual stimuli, spoken sentences, or the development of social interaction may unfold over very different time-scales. Thus, it is plausible to assume that the brain processes information in distinct frequency sub-bands. In the frequency-specific ISC analysis, we first filter the original time-series of each voxel (and subject) to multiple frequency sub-bands using an octave filter bank based on stationary wavelet transformation (SWT; Kauppi et al., 2010b). After band-pass filtering each fMRI time-series, we compute ISCs using the Equation (1) voxel-wise separately within each frequency sub-band and threshold the ISC maps using the same test as described in section 2.2.1.

It has been shown previously that wavelets are well-suited to analyze fMRI data because of certain properties of the cortical fMRI time-series, such as 1/f -like frequency characteristics (Bullmore et al., 2004). Moreover, the SWT algorithm is specifically suited to our analysis because it performs a time-invariant (Bradley, 2003) transformation unlike the discrete wavelet transform (DWT). In practice, this property means that a small difference in the hemodynamic delays of two fMRI time series transforms into a similar small difference in the filtered signals, allowing consistent estimation of the correlation coefficients between the subjects time-series after performing the filtering. For the DWT, even a minor delay between two identical input signals might cause a large difference in the filtered signals, making it much less-suited algorithm for frequency-specific ISC analysis. The SWT algorithm can be efficiently implemented using a sub-band coding scheme based on successive decimations of so called quadrature mirror filters (QMFs) and convolution operations (Vetterli and Kovačević, 1995).

We use Daubechies scaling and wavelet functions as a default filter option as they satisfy a necessary QMF relationship (Vetterli and Kovačević, 1995) and have been successfully applied to fMRI data earlier (Bullmore et al., 2001; Achard et al., 2006). The maximum degree of the polynomials the scaling function can reproduce is called the number of the vanishing moments. The number of Daubechies filter coefficients are associated with the number of the vanishing moments by the equation $K = 2V$, where K is the number of filter coefficients and V is the number of vanishing moments. We use short filters of length $K = 4$ as a default analysis option which are flexible enough to encode polynomials with two coefficients (both constant and linear signal components). In principle, the localization in the frequency domain could be improved by using higher filter lengths, but the use of long filters increases computation time (SWT needs to be computed separately for the time-series of every subject for each brain voxel) and makes the detection of rapid signal changes less accurate. In addition, because typical fMRI measurements

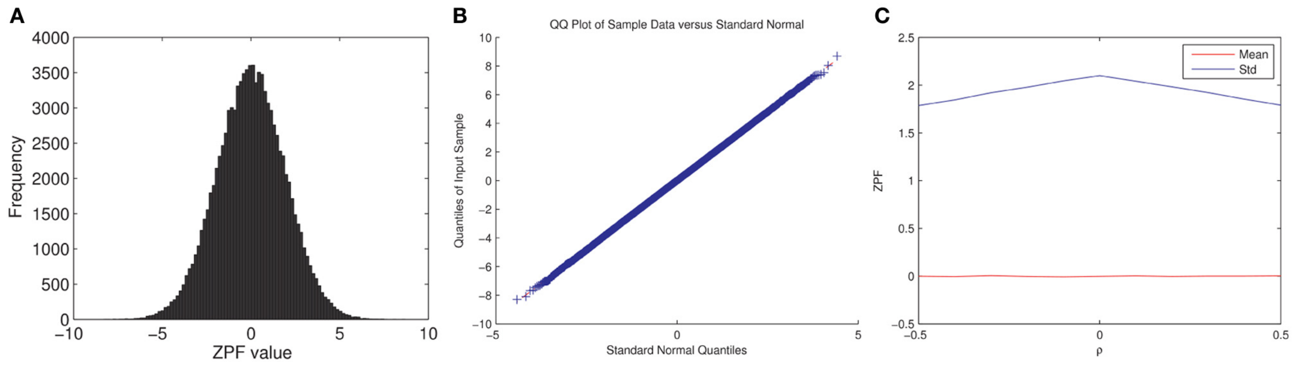


FIGURE 2 | ZPF Monte Carlo simulation. We simulated fMRI time series for two subjects and two conditions as $\mathbf{x}_s^a = \beta\mathbf{t}^a + \mathbf{n}_s^a$, where $\mathbf{x}_s^a(s,\ a = 1, 2)$ are the simulated time series, $\mathbf{t}^a$ is the common time-series between the two subjects in the condition a, $\mathbf{n}_s^a$ is pink noise [generated as in Pajula et al. (2012)], and β is selected so that the true correlation between the subjects' time-series is ρ, which is varied during the simulation. $\mathbf{t}^a$ was generated by smoothing white Gaussian noise with a box filter and convolving the resulting time series by a hemodynamic response function. The times series length was 100 with modeled TR of 2*s*. Note that ρ has the same value for both conditions, according to the null-hypothesis. The simulation does not model for the dependence between the two conditions. The simulation was repeated 100000 times for each $\rho = -0.5, -0.4, \ldots, 0.5$. **(A)** Shows the histogram of ZPF values when $\rho = 0.3$, **(B)** Shows the QQ-plot for the same case, and **(C)** Shows the average value and the standard deviation of the ZPF-statistics as a function of ρ, where a slight dependence of the standard deviation on ρ is observed. As it is visible (1) the distribution of the ZPF was Gaussian (with a larger variance than 1), and (2) its parameters did not markedly depend on the value of ρ.

contain relatively low number of time points, short filters are preferred to minimize boundary artifacts. Daubechies basis functions are optimal in the sense that they provide the shortest filter length for the given number of vanishing moments. However, also other basis functions have been proposed for fMRI data analysis. For instance, Ruttimann et al. (1998) used symmetric spline wavelets because of their phase-preserving property.

2.2.6. Time window ISC analysis

When analyzing complex fMRI data sets such as those collected during a movie watching, it is likely that ISCs vary drastically over the experiment. To analyze how ISC varies over time, it can be highly useful to compute ISC maps for several consecutive possible overlapping time windows. With the ISC toolbox, a user can specify suitable time window parameters (window length, step length between two consecutive windows) and compute "short-time ISC maps" for each window. To obtain these maps, we compute $\bar{r}$ statistic (across all voxels and subjects) within each time-window and assess the significance of the ISCs as described above. We randomize the generation of the null distribution across all time windows which leads to a common threshold for all windows. The length of the time window has to be sufficient to obtain reliable estimates of $\bar{r}$ for each time window. The choice depends on the number of subjects and the type of the stimulus. Therefore, it is not straight-forward to give exact suggestions about the minimal time-window length. However, window lengths as short as 10 samples have been used (Nummenmaa et al., 2012).

The toolbox allows the visualization of the time window ISC maps over an anatomical template. It also automatically computes the mean of $\bar{r}$-values across voxels within different brain region-of-interest (ROIs), allowing plotting ROI-averaged ISCs over time. These curves can be correlated with the features of the stimuli, behavioral ratings or other variables of interest.

2.2.7. Intersubject phase synchronization

Time window ISC and frequency-specific ISC analyses can provide neuroscientifically meaningful insights into complex fMRI data. An obvious way to combine benefits of both approaches is to compute frequency-specific ISC maps in several time windows to investigate temporal evolution of the ISCs in specific time-scales. ISC toolbox automatically computes also these maps if the user performs both time window ISC and frequency-specific ISC analyses. A limitation of this approach is that the temporal resolution of the analysis can be modest because each time window must contain several time points to allow meaningful interpretation of the correlation coefficient. This problem is most prominent in the lowest frequency sub-bands because the temporal resolution of slow fluctuations is inherently poor as stated by the time-frequency uncertainty principle (Cohen, 1995). To increase the temporal resolution of the time-varying analysis in distinct frequency sub-bands, we propose using phase synchronization between subjects as a measure of inter-subject similarity. A similarity measure based on instantaneous phase allows the analysis of the band-pass filtered signals on the basis of inherent temporal resolution of the time series. This is in contrast to the time window ISC analysis for which the resolution is further limited by the length of the time-window.

Many phase synchronization measures have been designed to analyze functional neuroimaging signals (Vinck et al., 2011) but they are mainly used to analyze electroencephalography and magnetoencephalography signals. Unlike these signals, fMRI time-series may not be characterized by oscillatory activity. However, the analysis of the instantaneous phases still remains a valid method to characterize a specific interrelation between phases (Pikovsky et al., 2000; Laird et al., 2002). To extract phase information, complex-valued analytic time-series must be available. Hence, we apply the Hilbert transform (Goswami and Hoefel, 2004) to the fMRI time-series to obtain their corresponding analytic signals[3]. We take the *absolute angular distance* (Vinck et al., 2010) between the time-series of two subjects as a dissimilarity measure:

$$p_{ij}(t) = |\theta_i(t) - \theta_j(t)| \mod \pi, \tag{4}$$

where t is a time-point index and angles θ_i, θ_j are computed based on the analytical time-series measured from subjects i and j. This is an intuitive measure of phase interrelationship between the fMRI time-series of two subjects: If fluctuations of the (band-limited) time-series between subjects are highly similar, it is expected that the absolute phase difference is smaller than when fluctuations are different. There are different possibilities to extend this measure to group-level analysis (Glerean et al., 2012). We use a comparable definition to our ISC measure [Equation (1)] and compute the average of all subject-pairwise absolute angular distances as:

$$\overline{p}(t) = \frac{1}{N(N-1)/2} \sum_{i=1}^{N} \sum_{j=2,\ j>i}^{N-1} p_{ij}(t). \tag{5}$$

Our final measure of inter-subject phase synchronization (IPS) is the normalized version of $\overline{p}$:

$$\hat{\overline{p}}(t) = 1 - \frac{\overline{p}(t)}{\pi}. \tag{6}$$

This measure has its values always within the range [0 1], where the value 1 indicates a complete phase similarity and the value 0 corresponds to a complete *absence* of phase similarity across subjects. Similarly to time window ISCs, the ISC Toolbox allows different plotting options for IPS results. For instance, averaged IPS values within selected ROIs can be plotted over time. These curves can be then correlated with the features of the stimuli or other variables of interest.

2.3. IMPLEMENTATION

As explained in section 2.1, the use of the ISC Toolbox starts from the startup GUI where a user defines requested analyses and their parameters (see **Figure 1**). The GUI automatically detects the operating system and checks that all necessary software and files are available. After a user has selected desired analysis options, the GUI validates them. After a successful validation, the parameters are set in a structure array called *Params* which is saved in a mat-file. The GUI also generates the destination directory and all necessary sub-directories for the analysis results.

The computational analysis is controlled inside the main function named *runAnalysis*. The Matlab code of this function is grouped in six computational stages to clarify how the computations can be distributed across a computer cluster:

Stage 1 Binary data files for the analysis results as well as the memory map pointers to access these files are initialized. The pointers are saved in the structure called *memMaps* which is saved in the analysis destination directory. In the later stages of the program, the files are repeatedly accessed and modified using these pointers (see more information about the Matlab's memory mapping feature below).

Stage 2 The wavelet filtering for the frequency-specific ISC analysis is performed.

Stage 3 Average ISC maps are computed, including the generation of the re-sampling distributions for the assessment of statistical thresholds.

Stage 4 Critical thresholds are calculated based on the re-sampling distributions including threshold correction for multiple comparisons. In the FDR-based correction, p-values for statistically significant (before a multiple comparison correction) samples need to be available. These are estimated in a nonparametric fashion from the observations of the re-sampling distribution using a linear interpolation.

Stage 5 Inter-subject synchronization curves over time are computed for the time window ISC and IPS for all the brain regions and thresholds defined in the Harvard-Oxford sub-cortical and cortical atlases.

Stage 6 All the generated statistical maps in the previous stages are saved to the analysis destination folder as nifti files. This stage is always computed locally even if a grid enviroment would be available.

The grouping of the code is based on the dependencies of the analysis pipeline: the execution of the functions within any of the stages is always dependent on the results of the preceding stage and therefore cannot be performed before all previous stages have been completed and their intermediate results have been saved to the analysis destination directories. However, computations inside the loop structures *within* each computational stage are independent of each other, meaning that functions repeatedly called inside these loops can be equally well run in parallel. In practice, a user does not need to understand how the code is written because the program can automatically parallelize computations across a computer grid/cluster.

Only those stages corresponding to ISC based analyses that are requested by the user are run when executing *runAnalysis*. For example, if the frequency-specific ISC analysis is not chosen by a user, the stage 2 is skipped.

Matlab's memory mapping is a mechanism that maps a portion of a file, or an entire file, on disk to a range of addresses within an application's address space. The application can then access files on disk in the same way it accesses dynamic memory (The Mathworks Inc., 2013). This memory mapping mechanism is employed in the ISC Toolbox for three main reasons:

1. Because of a large memory demand, all the data cannot be held in the central memory all the time.
2. The traditional file I/O can be very slow especially in cluster computing environments.

[3] The analytic signal $\mathbf{x}_a(t) = x(t) + jy(t) = A(t)e^{j\theta(t)}$ can represent both the instantaneous amplitude envelope $A(t)$ and phase $\theta(t)$ of the time-series, but only phase information $\theta(t) = \arctan\left(\frac{y(t)}{x(t)}\right)$ is used to derive our phase similarity measure.

3. The memory mapping provides a mechanism for sharing the memory between multiple processes that is important for the cluster computing abilities in the ISC Toolbox.

The disadvantage of the used memory mapping mechanism is that it is highly hardware and also somewhat operating system and Matlab version dependent. The memory mapped data can become corrupt or unreadable if the used hardware or the Matlab version is changed. In the ISC Toolbox, the problem is circumvented by saving the important results out from the memory maps to nifti files. The corrected statistical thresholds are saved as Matlab's mat-file and also as a text file. Therefore, the visualization of the thresholded maps can be done afterwards easily with any visualization software. The memory mapping has been previously used in the SurfStat software within brain imaging (Worsley, 2008).

A heavy computational burden is one of the major issues when using the ISC Toolbox. Computations require large memory as mentioned already and they also take a long time to compute. Currently, the ISC Toolbox supports cluster computing in SGE-based (Oracle Grid Engine, Son of a Grid Engine, or Open Grid Scheduler) and Slurm (Simple Linux Utility for Resource Management) environments. Generally, the SGE based parallelization (Love, 2013; Scalable Logic, 2013; Univa Corporation, 2013) has been used extensively within brain imaging software such as FSL. The Slurm grid engine (GE) (Yoo et al., 2003) is currently becoming more common and for this reason also Slurm based parallelization was selected to be supported in the ISC Toolbox. The only requirements to use parallelization procedures in the toolbox are that the operating system of the used computer must be Linux and the user must have access to system running on one of these two GEs.

In both cases (SGE or Slurm), separate shell scripts must be generated for each computational stage before distributing them to the GE. The script generation and submission to GE is handled with the function *gridParser*. The *gridParser* function generates separate shell scripts for each process stage of each possible parallel process and submits these to the current GE. Simplified examples from the shell scripts generated by *gridParser* for the first stage of execution are presented in Listings 2.3 and 2.3. In this example, the project name is "ISC_test_analysis," which defines the mat-file name for the Params struct. *memMapData* function implements the stage 1 of the analysis. The only input for the function is the Params struct. The number of generated scripts varies from 4, for the basic analysis using a single CPU, to hundreds depending on the selected analyses and the degree of parallelization.

Listing 1. Bash script example for the Stage 1 of the analysis generated by the *gridParser* function for the SGE environment.

```
matlab –nosplash –nodisplay –nojvm –n
  odesktop –r "addpath(genpath('/home
  /testuser/ISCofficial/isc-toolbox/'));
  load('/home/testuser/ISCtest/ISC_test
  _analysis'); memMapData(Params); exit"
exit
```

Listing 2. Bash script example for the Stage 1 of the analysis generated by the *gridParser* function for the Slurm environment.

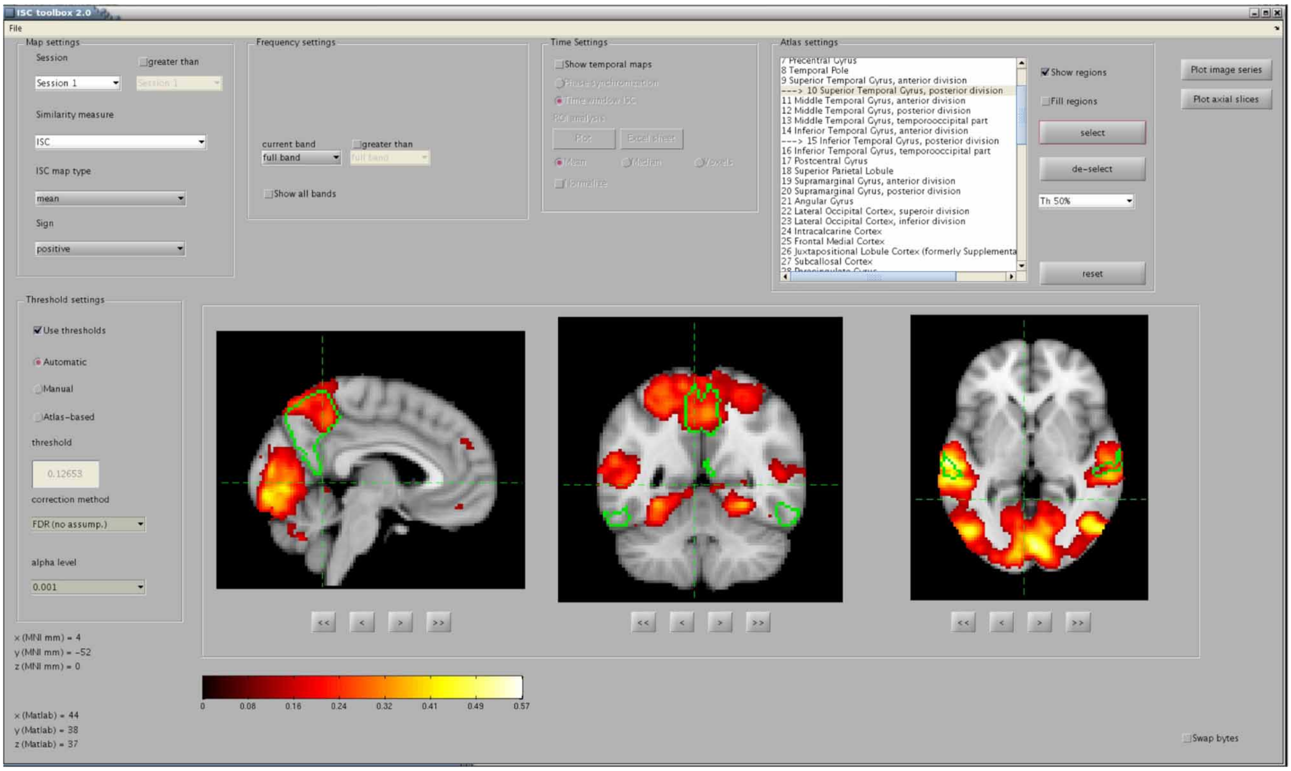
```
#!/bin/sh
module load matlab
matlab –nosplash –nodisplay –nojvm –n
  odesktop –r "addpath(genpath('/home
  /testuser/ISCofficial/isc-toolbox/'));
  load('/home/testuser/ISCtest/ISC_test
  _analysis'); memMapData(Params); exit"
exit
```

The monitoring of the submitted tasks is handled with the function *waitGrid*. The function requests the running processes in the GE in defined time interval and prevents the main function to continue before all submitted sub-processes are finished.

The data integrity is always a critical question within parallel computing. It must be ensured that any two processes are not interfering each other and all data are saved safely. The function *freeToWrite* was developed to maintain the data integrity. It handles a specific lock system to ensure that only one process updates the memory maps at once. The lock is based on a simple lock-file which is generated before the data are going to be saved and deleted when the saving process has been finished. Every process which updates the memory maps are using the lock system. To simplify the debugging, every lock file has its own identifier based on the name of the process and the current process ID from the GE.

2.4. VISUALIZATION GUI

To simplify the investigation of the ISC analysis results a separate visualization GUI, shown in **Figure 3**, was developed to interactively show the statistics maps and other results computed by the ISC Toolbox. The visualization GUI can show all the statistical maps resulting from the analyses by the toolbox. There are several software tools for high quality, interactive visualizations of the statistical maps from neuroimaging analyses. However, as far as we know, none of these is suitable for the visualization of advanced ISC analysis such as 4-D statistical maps of the time window ISC. In addition, a specialized visualization application provides additional convenience by allowing user to switch between different analysis results by a quick button press instead of a cumbersome reloading of the statistical maps one-by-one from a disk. We avoid the re-loading of the statistical maps by directly accessing the data portion of interest from the disk. The fast random access to the data is possible because the ISC results were mapped to a disk during the main analysis procedure with the aid of memory-mapping. Most of the data which are presented or used for creating the visualizations in the GUI are precomputed by the main analysis procedure and mapped to a memory. The memory-mapping minimizes the need of RAM, which enables the efficient interactive visualization and exploration of the analysis results also with slower computers. A price to pay for this added flexibility are possible cross-platform incompatibility issues, mentioned already in section 2.3, if the actual analysis is carried out with a different hardware than with which the analysis results are viewed. The simplest of these issues is the endianness, which can be changed by ticking the checkbox "Swap bytes."



ISC toolbox 2.0

File

Map settings

Session: Session 1 ☐ greater than Session 1

Similarity measure: ISC

ISC map type: mean

Sign: positive

Frequency settings

current band: full band ☐ greater than full band

☐ Show all bands

Time Settings

☐ Show temporal maps

○ Phase synchronization

● Time window ISC

ROI analysis

○ ● Mean ○ Median ○ Voxels

☐ Normalize

Atlas settings

7 Precentral Gyrus
8 Temporal Pole
9 Superior Temporal Gyrus, anterior division
---> 10 Superior Temporal Gyrus, posterior division
11 Middle Temporal Gyrus, anterior division
12 Middle Temporal Gyrus, posterior division
13 Middle Temporal Gyrus, temporooccipital part
14 Inferior Temporal Gyrus, anterior division
---> 15 Inferior Temporal Gyrus, posterior division
16 Inferior Temporal Gyrus, temporooccipital part
17 Postcentral Gyrus
18 Superior Parietal Lobule
19 Supramarginal Gyrus, anterior division
20 Supramarginal Gyrus, posterior division
21 Angular Gyrus
22 Lateral Occipital Cortex, superoir division
23 Lateral Occipital Cortex, inferior division
24 Intracalcarine Cortex
25 Frontal Medial Cortex
26 Juxtapositional Lobule Cortex (formerly Supplementa
27 Subcallosal Cortex

☑ Show regions

☐ Fill regions

Th 50%

Threshold settings

☑ Use thresholds

● Automatic

○ Manual

○ Atlas-based

threshold: 0.12653

correction method: FDR (no assump.)

alpha level: 0.001

x (MNI mm) = 4
y (MNI mm) = -52
z (MNI mm) = 0

x (Matlab) = 44
y (Matlab) = 38
z (Matlab) = 37

☐ Swap bytes

FIGURE 3 | The main window of the visualization GUI. In the shown analysis example, a user has located significant ISCs in several brain areas, including the precuneus cortex and the posterior division of the superior temporal gyrus, whose perimetries are shown in green color over the anatomical template. The map was thresholded and FDR corrected (q <0.001) over the whole brain using the re-sampling test of section 2.2.1.

In addition to minimizing the memory consumption and the access time to the data in a disk, it is important to minimize the time that is spent for plotting accessed data on the screen. In Matlab, the plotting of the images is much faster when using indexed images (in an integer format) than using true color images or intensity images in a floating point format. Indexed images are fast to visualize because they use direct mapping of pixel values to colormap values. Hence, to maximize the browsing speed, the GUI converts ISC maps and anatomical templates from a floating point format to an integer format and combines these data into a single matrix of integer values. An appropriate colormap is then created to allow visualization of the indexed image on the screen in multiple colors. The colormap involves hot (yellow and red), cold (magenta and blue), and gray colors which allows the visualization of positive and negative ISCs as well as anatomical intensity values over a single image.

The exact appearance of the Visualization GUI on the screen depends which analyses, described in section 2.2, the user has run. For example, if only the basic ISC analysis has been run the visualization GUI enables only the analysis of ISCs across a whole session and frequency-spectrum by disabling “temporal settings” and “frequency settings” -panels. Statistical maps are shown in sagittal, coronal and axial views. The MNI coordinates of the views can be changed via the buttons below the axis. An additional option is to visualize several axial slices across the whole brain volume in a single figure. The exploration of the volume along a fourth dimension (time interval or frequency range) currently requires a button press. A user can also select Harvard-Oxford probabilistic atlas regions for the visualization over the statistical map and it is possible to view average ISCs for selected ROIs as a function of time. In addition to these visualizations, the GUI contains more advanced visualization options which allow detailed localization of ISCs in spatial, temporal and spectral dimensions.

The GUI allows fast and comprehensive visualization of the ISC analysis results in an exploratory manner. However, to address specific research questions, a further analyses not supported by the ISC Toolbox may be needed. Moreover, it may also be meaningful to customize the way how the results are visualized. For these purposes, the GUI has an option to export ISC maps and other results to the Matlab’s workspace as variables. Although the ISC analysis results are also saved in a disk as Nifti-files and are freely accessible for a user, the export option allows quick and easy visualization of the threholded maps over an anatomical image and selected atlas regions for the dimensions (spatial, temporal, and spectral) of interest.

3. APPLICATIONS

Next, we shortly exemplify how the toolbox has been successfully used to analyze fMRI data.

3.1. BASIC ISC ANALYSIS FOR ACTIVATION DETECTION

A primary interest in many fMRI based imaging studies is to detect brain locations associated with a task related neural activity. Traditionally, this is achieved by a GLM based analysis, where voxel time courses are compared to the task-derived reference time course. The application of the GLM requires explicit knowledge how stimuli are varied during the experiment and cannot therefore be used to detect activations from experiments involving complex naturalistic stimuli. Pajula et al. (2012) showed that our "basic" ISC analysis described in section 2.2.1 is a suitable method to detect task related neural activation without making any assumptions about the applied stimuli. In this study, fMRI data from 37 right-handed subjects who all had performed the same five blocked design tasks [4] were analyzed with both ISC Toolbox and a GLM based method. The idea is that the GLM-detected activations with this kind of strictly controlled and well-known tasks can be assumed to be reliable and can be treated as a gold standard. Interestingly, the comparison of the statistical maps of ISC and GLM revealed high agreement of the findings. This demonstrates that the ISC analysis can detect truly active brain regions in a manner that is completely "blind" to stimuli, making it highly promising method for detecting activity in data sets collected under naturalistic stimuli experiments.

3.2. ISC DIFFERENCE MAPS FOR ANALYSIS OF AESTHETIC EXPERIENCES

Understanding how spectators' brains process information during an aesthetic performance, such as a dance performance, is an interesting topic in neuroscience. To investigate this, videos of aesthetic performances can be shown to subjects while their brain activity is being measured using the fMRI. Stimuli in these experiments are very rich, making ISC based methods a natural choice for data analysis. Reason et al. (under review) used ISC toolbox to study whether auditory stimulation have an effect on the kinesthetic experience and/or the aesthetic appreciation of the spectator while watching dance. In the study, fMRI signals were acquired from 22 subjects under two different stimulus conditions: (1) a full audiovisual dance performance accompanied by the soundscapes of Bach (condition = "Bach"), and (2) the same dance performance without the music, including only visual stimuli as well as sounds of breathing and footfalls of the dancer (condition = "Breathing"). ISC toolbox was used to construct individual ISC maps of both conditions as described in section 2.2.1 as well as to construct ISC difference maps "Bach" < "Breathing" and "Bach" > "Breathing" as described in section 2.2.4.

The individual ISC maps showed large overlap in the visual and auditory cortices for both conditions. However, the analysis of the ISC difference maps revealed clusters in the temporal cortex that were unique to the different audio conditions, indicating also clear differences between the processing of the sound in the "Bach" and "Breathing" conditions. Based on detailed investigation of the ISC difference maps, Reason et al. (under review) suggested several possibilities how the presence or absence of music may influence spectators' experience. For instance, the postcentral gyrus of parietal cortex (BA 7) showed significantly greater ISC in the "Breathing" condition. The area is known for simultaneously processing multiple sensory modalities, in particular the somesthetic modality that includes touch. This somesthetic connection implies a form of motor cognition and could suggest that the "Breathing" elicited greater engagement of action understanding within body-specific mechanisms.

3.3. FREQUENCY-SPECIFIC ISC FOR ANALYSIS OF TEMPORAL BRAIN HIERARCHY

In our previous study (Kauppi et al., 2010b), we performed frequency-specific ISC analysis to investigate processing of movie events that occur over multiple time-scales. We analyzed fMRI data collected from the experiment (Jääskeläinen et al., 2008) where 12 subjects watched the 36 min clip of an Academy Award winning drama movie Crash (Lions Gate Films, 2005, directed by Paul Haggis; the movie was presented with sound). We constructed both frequency-specific ISC maps described in section 2.2.5 as well as ISC difference maps to compare differences in ISCs between distinct frequency subbands (see section 2.2.4).

The frequency-specific ISC analysis provided novel and interesting insights into the highly complex fMRI data. For instance, the analysis revealed that visual cortical ISC was present across the whole frequency spectrum of the fMRI signal, ISC in temporal areas occurred in all but the highest frequency band, and frontal cortical ISC was present only in the two lowest frequency bands. Hence, the frequency range showing significant ISC *contracted* when moving from lower-order sensory areas toward higher-order cortical areas. There are several possible explanations for the mappings found in this study. For instance, the findings might reflect the hierarchy of temporal receptive windows (TRWs) in the human brain, with sensory visual cortical areas showing short TRWs, and the TRWs becoming progressively longer as one ascends to functionally higher-order cortical areas (Hasson et al., 2008).

3.4. TIME WINDOW ISC FOR ANALYSIS OF HIGHER-ORDER BRAIN FUNCTIONS

The use of movies as stimuli in neuroimaging studies offers new possibilities to understand higher-order brain functions, such as those related to social cognition and emotions. Nummenmaa et al. (2012) used the time window ISC (which they call moment-to-moment ISC) to analyze how ISC is associated with events that elicit emotions in movies. Functional MRI data from 16 subjects were collected while they watched movies depicting unpleasant, neutral, and pleasant emotions. After scanning, participants watched the movies again and continuously rated their experience of pleasantness–unpleasantness (i.e., valence) and of arousal–calmness. Short-time ISCs for each voxel were then computed using the ISC toolbox as described in section 2.2.6, using a 17-s sliding window (a step size of the time-window was one time point). Time series of valence and arousal ratings were then used to predict temporal variation of ISCs within each voxel.

Negative valence was associated with increased ISC in the emotion-processing network (thalamus, ventral striatum, insula) and in the default-mode network (precuneus, temporoparietal junction, medial prefrontal cortex, posterior superior temporal sulcus). High arousal was associated with increased ISC in the somatosensory cortices and visual and dorsal attention networks

[4] Functional MRI data from the measurements with Functional Reference Battery tasks developed by the International Consortium for Human Brain Mapping (ICBM) were used (Mazziotta et al., 2001): http://www.loni.ucla.edu/ICBM/Downloads/Downloads_FRB.shtml.

comprising the visual cortex, bilateral intraparietal sulci, and frontal eye fields. It was proposed that negative valence synchronizes individuals brain areas supporting emotional sensations and understanding of anothers actions, whereas high arousal directs individuals attention to similar features of the environment.

3.5. IPS FOR TIME-VARYING ANALYSIS OF NATURALISTIC fMRI DATA

IPS described in section 2.2.7 is an alternative option for time window ISC to analyze complex fMRI data over time. Glerean et al. (2012) applied both time window ISC and IPS analysis for naturalistic fMRI data collected from 12 subjects while they watched a feature movie (for details of the experiment, see Lahnakoski et al., 2012). IPS was computed within a frequency-band of 0.04–0.07 Hz and a time window ISC was computed for several window sizes from 4 to 32 samples (corresponding to window lengths from 8 to 64-s with the TR of 2-s) using a sliding window.

A major conclusion of the study was that the IPS approach provided improved temporal resolution as compared with the time window ISC. In addition, an anatomical mapping of the whole-brain temporal average of the IPS was highly consistent with the anatomical mapping of the ISC computed across the whole movie experiment (without using time windows), indicating that the IPS is a realiable measure of inter-subject similarity.

4. COMPUTATION TIME

The computation time was measured in three different hardware setups utilizing the both the local and distributed computing abilities of the ISC Toolbox. The local computations were tested with Dell Optiplex 755 desktop computer equipped with Intel Core2Duo E8400 CPU @ 3.00 GHz and 5GB read access memory (RAM). The distributed computations were tested in two computing clusters. The larger cluster, called Merope, had nodes running on HP ProLiant SL390s G7 equipped with Intel Xeon X5650 CPU 2,67 GHz and minimum of 4 GB RAM / core. The GE was Slurm. The smaller of the tested computing clusters, called Outolintu, was running with SGE and had nodes running on IBM System x3550 equipped with two Intel Xeon X5450 CPUs 3.0 GHz and 32 GB RAM (with 10 GB swap) for each node.

In the Merope cluster, on average 32 processes were run simultaneously. With the Outolintu cluster the maximum of parallel processes was limited to 10 due to global usage limitations for a single user of this cluster. The computing times of cluster environments were averaged from three separated runs as in the cluster the computing time can be affected from the current load of the cluster as well as the implementation of the distributing system causes a small variation on computing time.

The computing time was measured from "the user perspective": Starting from the moment when user pushes the "Run Analysis" -button of the startup GUI to the moment when the analysis was finished. In a cluster environment, this means that the processing times of the GE were included to the total processing time.

The analyses were performed for the same measurement data which was used in earlier studies with the ISC Toolbox (Jääskeläinen et al., 2008; Kauppi et al., 2010b). The data was acquired from 12 subjects ($TR = 3.4$ s, 244 time points) and was registered to MNI152 space (for details see Kauppi et al., 2010b). The image dimensions were 91 × 109 × 91 × 244 (X × Y × Z × time) which resulted in an 840 MB file size for each subject and 9.8 GB total size of the analysis data set.

The tested ISC Toolbox setups were "basic ISC", "basic ISC + time window ISC," and "basic ISC + frequency-specific ISC." The first setup computed the ISC map across the entire length of the time-series and constructed a re-sampling distribution based on 100 million random shufflings of the time-series as in our earlier study (Kauppi et al., 2010b). The second setup was similar to the first setup except that the time window ISC with the window length and window step of 30 samples was used in addition to the basic ISC analysis. The third setup used frequency-specific ISC with three frequency sub-bands instead of time window ISC. The number of randomizations to threshold the ISC $\bar{r}$-maps was the same as in setups 1 and 2, and 25,000 random permutations for each brain voxel was used to construct a null permutation distribution of the sum ZPF statistic to allow thresholding of ISC difference maps between frequency bands.

The computing times are presented in **Figure 4**. On a single desktop computer, the computing time varied from 11 h 52 min to 25 h 20 min depending from the selected analysis. On the smaller Outolintu cluster, the corresponding times varied from 1 h 24 min to 6 h 10 min and, on the larger Merope cluster from 33 min to 3 h 25 min. Comparing local and distributed systems, the speed up factor was 10 with Outolintu and 24 with Merope in the first two setups. For the final setup with the frequency band analysis, the speed up factor was 4 with Outlintu and 7.5 with Merope.

The smaller speed up factors for the frequency band analysis was probably due to a higher number of hard drive interactions involved in this analysis as compared with the other tested analyses. In a cluster computing environment, a high number of hard drive interactions slows down the computations as the data is commonly located on a network drive and the speed of the data transfer in a network is usually clearly slower than the speed of data transfer via the internal bus of a desktop computer.

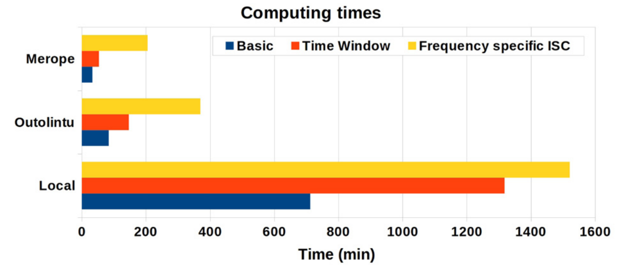


FIGURE 4 | The computing times from desktop computer and two cluster environments. The desktop computer was equipped with Intel Core2Duo E8400 CPU 3.00 GHz and 5 GB RAM. Ten parallel processes were run on Outolintu cluster with nodes equipped with Intel Xeon X5450 CPUs 3.0 GHz. On average, 32 parallel processes were run on Merope cluster with nodes equipped with Intel Xeon X5650 CPUs 2,67 GHz.

5. DISCUSSION

Branches of neuroscience investigating brain functions in experiments mimicking real-world conditions are growing rapidly and the development of data analysis methods must address an increasing diversity of research questions. The ISC based approaches can address key questions such as how processing differs between two groups (e.g., healthy vs. nonhealthy) exposed to identical complex stimuli or between two conditions (e.g., silent vs. nonsilent video). The new analyses methods incorporated in the ISC toolbox, described in sections 2.2.4–2.2.7, are one of the first attempts to help neuroscientists to address these and other aspects of neural processing. In addition to these new features, the toolbox will be continuously updated in the future to allow even more versatile analyses. For instance, the toolbox is currently limited to analyze between-subject correlations in a voxel-wise manner and does not allow more general investigations of functional correspondence between spatially disjoint brain areas across subjects. Features to analyze ISCs between different brain areas both within- and across subjects will be incorporated in the future versions of the toolbox.

One limitation of the current analysis approach is that the used ISC measure does not capture any information about the variability of the ISCs among subjects as it is simply the average of the upper-triangular (or lower-triangular) elements of the between-subject correlation matrix computed separately for each voxel [Equation (1)]. The consequence of the averaging is that interesting features of brain processing may be missed especially in higher-order brain regions where inter-subject variability is expected to be very high. To increase the sensitivity of the existing method to localize interesting brain areas as well as to perform more fine-grained ISC based analyses, it can be highly useful to preserve and analyze the entire structure of the between-subject correlation matrices. We have already taken steps toward this direction (Kauppi et al., 2010a) and will equip the toolbox with matrix-based analysis methods in the future.

One of the key issues in ISC based analyses is how to select a suitable threshold to distinguish meaningful ISC values from spurious ones. Because of the restrictive assumptions made by standard parametric statistical procedures, such as the ordinary *t*-test, we have decided to use fully nonparametric re-sampling based methods to determine the critical thresholds to improve reliability of the analysis. Despite of the flexibility of the nonparametric methods, it is important to keep in mind that also they provide only approximations of true, underlying null re-sampling distributions. This is due to finite number of realizations drawn as well as certain assumptions required by the tests which may not be fulfilled by real fMRI time-series. However, as shown by the results, our easy and fully automated mechanism which distributes calculations across a computational cluster allows drawing huge number of realizations in a relatively short time, making the generation of accurate re-samplings distributions feasible. Moreover, we showed with a simple Monte-Carlo simulation that certain critical assumptions made by the sum ZPF test are not violated in practice. In any case, further validation and improvement of our current statistical procedures is another important topic of future research.

6. CONCLUSIONS

We have presented a software package, named ISC Toolbox, implemented in Matlab for computing various ISC based analyses. The computations can be launched from a GUI making the use of the toolbox easy. Many advanced techniques such as time window ISC analysis, frequency-specific ISC analysis, IPS analysis and the comparison of ISCs between different stimuli are supported by the toolbox. The analyses are coupled with nonparametric re-sampling based statistical inference methods. As these analyses are computationally intensive, the ISC Toolbox is equipped with automated cluster computing mechanisms to reduce the computation time via parallelization and a marked reduction in computation time was achieved by cluster computing. The ISC Toolbox is available in https://code.google.com/p/isc-toolbox/ under the MIT open source licence.

ACKNOWLEDGMENTS

FUNDING

This research has been supported by the Academy of Finland under the grants 130275, 263785 to Jussi Tohka and under the National Centers of Excellence Programme in Algorithmic Data Analysis, and Computational Sciences Program.

REFERENCES

Abrams, D. A., Ryali, S., Chen, T., Chordia, P., Khouzam, A., Levitin, D. J., et al. (2013). Inter-subject synchronization of brain responses during natural music listening. *Eur. J. Neurosci.* 37, 1458–1469. doi: 10.1111/ejn.12173

Achard, S., Salvador, R., Whitcher, B., Suckling, J., and Bullmore, E. (2006). A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *J. Neurosci.* 26, 63–72. doi: 10.1523/JNEUROSCI.3874-05.2006

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B* 57, 289–300.

Bradley, A. P. (2003). "Shift-invariance in the discrete wavelet transform," in *Proceedings of VIIth Digital Image Computing: Techniques and Applications* (Sydney).

Bullmore, E., Fadili, J., Maxim, V., Şendur, L., Whitcher, B., Suckling, J., et al. (2004). Wavelets and functional magnetic resonance imaging of the human brain. *Neuroimage* 23, S234–S249. doi: 10.1016/j.neuroimage.2004.07.012

Bullmore, E., Long, C., Suckling, J., Fadili, J., Calvert, G., Zelaya, F., et al. (2001). Colored noise and computational inference in neurophysiological (fMRI) time series analysis: resampling methods in time and wavelet domains. *Hum. Brain Mapp.* 12, 61–78. doi: 10.1002/1097-0193(200102)12:2<61::AID-HBM1004>3.0.CO;2-W

Cohen, L. (1995). *Time-Frequency Analysis*, Vol. 778. Englewood Cliffs, NJ: Prentice Hall PTR.

Glerean, E., Salmi, J., Lahnakoski, J. M., Jääskeläinen, I. P., and Sams, M. (2012). Functional magnetic resonance imaging phase synchronization as a measure of dynamic functional connectivity. *Brain Connect.* 2, 91–101. doi: 10.1089/brain.2011.0068

Good, P. (2005). *Permutation, Parametric and Bootstrap Tests of Hypotheses.* 3rd Edn. New York, NY: Springer.

Goswami, J. C., and Hoefel, A. E. (2004). Algorithms for estimating instantaneous frequency. *Signal Process.* 84, 1423–1427. doi: 10.1016/j.sigpro.2004.05.016

Hasson, U., Nir, Y., Levy, I., Fuhrmann, G., and Malach, R. (2004). Intersubject synchronization of cortical activity during natural vision. *Science* 303, 1634–1640. doi: 10.1126/science.1089506

Hasson, U., Yang, E., Vallines, I., Heeger, D. J., and Rubin, N. (2008). A hierarchy of temporal receptive windows in human cortex. *J. Neurosci.* 28, 2539–2550. doi: 10.1523/JNEUROSCI.5487-07.2008

Jääskeläinen, I. P., Koskentalo, K., Balk, M. H., Autti, T., Kauramäki, J., Pomren, C., et al. (2008). Inter-subject synchronization of prefrontal cortex hemodynamic activity during natural viewing. *Open Neuroimag. J.* 2, 14–19. doi: 10.2174/1874440000802010014

Jola, C., McAleer, P., Grosbras, M.-H., Love, S. A., Morison, G., and Pollick, F. E. (2013). Uni-and multisensory brain areas are synchronised across spectators when watching unedited dance recordings. *Iperception* 4, 265–284. doi: 10.1068/i0536

Kauppi, J., Jääskeläinen, I., Sams, M., and Tohka, J. (2010a). "Clustering intersubject correlation matrices in functional magnetic resonance imaging," in *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on* (Corfu), 1–6.

Kauppi, J.-P., Jääskeläinen, I. P., Sams, M., and Tohka, J. (2010b). Inter-subject correlation of brain hemodynamic responses during watching a movie: localization in space and frequency. *Front. Neuroinform.* 4:5. doi: 10.3389/fninf.2010.00005

Kauppi, J.-P., Huttunen, H., Korkala, H., Jääskeläinen, I. P., Sams, M., and Tohka, J. (2011). "Face prediction from fMRI data during movie stimulus: strategies for feature selection," in *Artificial Neural Networks and Machine Learning–ICANN 2011,* eds T. Honkela, W. Duch, M. Girolami, and S. Kaski (Espoo: Springer), 189–196. doi: 10.1007/978-3-642-21738-8_25

Krishnamoorthy, K., and Xia, Y. (2007). Inferences on correlation coefficients: one-sample, independent and correlated cases. *J. Stat. Plan. Infer.* 137, 2362–2379. doi: 10.1016/j.jspi.2006.08.002

Lahnakoski, J. M., Glerean, E., Salmi, J., Jääskeläinen, I. P., Sams, M., Hari, R., et al. (2012). Naturalistic fMRI mapping reveals superior temporal sulcus as the hub for the distributed brain network for social perception. *Front. Hum. Neurosci.* 6:233. doi: 10.3389/fnhum.2012.00233

Laird, A. R., Rogers, B. P., Carew, J. D., Arfanakis, K., Moritz, C. H., and Meyerand, M. E. (2002). Characterizing instantaneous phase relationships in whole-brain fMRI activation data. *Hum. Brain Mapp.* 16, 71–80. doi: 10.1002/hbm.10027

Lerner, Y., Honey, C. J., Silbert, L. J., and Hasson, U. (2011). Topographic mapping of a hierarchy of temporal receptive windows using a narrated story. *J. Neurosci.* 31, 2906–2915. doi: 10.1523/JNEUROSCI.3684-10.2011

Love, D. (2013). *Son of Grid Engine Project*. Available online at: https://arc.liv.ac.uk/trac/SGE (Visited 5.11.2013).

Mazziotta, J., Toga, A., Evans, A., Fox, P., Lancaster, J., Zilles, K., et al. (2001). A probabilistic atlas and reference system for the human brain: International consortium for brain mapping (ICBM). *Philos. Trans. R Soc. Lond. B Biol. Sci.* 356, 1293–1322. doi: 10.1098/rstb.2001.0915

Nichols, T. E., and Holmes, A. P. (2002). Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Hum. Brain Mapp.* 15, 1–25. doi: 10.1002/hbm.1058

Nummenmaa, L., Glerean, E., Viinikainen, M., Jääskeläinen, I. P., Hari, R., and Sams, M. (2012). Emotions promote social interaction by synchronizing brain activity across individuals. *Proc. Natl. Acad. Sci. U.S.A.* 109, 9599–9604. doi: 10.1073/pnas.1206095109

Pajula, J., Kauppi, J.-P., and Tohka, J. (2012). Inter-subject correlation in fMRI: Method validation against stimulus-model based analysis. *PLoS ONE* 7:e41196. doi: 10.1371/journal.pone.0041196

Pikovsky, A., Rosenblum, M., and Kurths, J. (2000). Phase synchronization in regular and chaotic systems. *Int. J. Bifurcat. Chaos* 10, 2291–2305. doi: 10.1142/S0218127400001481

Raghunathan, T., Rosenthal, R., and Rubin, D. B. (1996). Comparing correlated but nonoverlapping correlations. *Psychol. Methods* 1:178. doi: 10.1037/1082-989X.1.2.178

Ruttimann, U. E., Unser, M., Rawlings, R. R., Rio, D., Ramsey, N. F., Mattay, V. S., et al. (1998). Statistical analysis of functional MRI data in the wavelet domain. *IEEE Trans. Med. Imaging* 17, 142–154. doi: 10.1109/42.700727

Scalable Logic. (2013). *Open Grid Scheduler/Grid Engine*. Available online at: http://gridscheduler.sourceforge.net/ (Visited 5.11.2013).

Schmälzle, R., Häcker, F., Renner, B., Honey, C., and Schupp, H. (2013). Neural correlates of risk perception during real-life risk communication. *J. Neurosci.* 33, 10340–10347. doi: 10.1523/JNEUROSCI.5323-12.2013

Spiers, H., and Maguire, E. (2007). Decoding human brain activity during real-world experiences. *Trends Cogn. Sci.* 11, 356–365. doi: 10.1016/j.tics.2007.06.002

The Mathworks Inc. (2013). *Overview of Memory-Mapping.* Matlab R2013b Documentation.

Univa Corporation. (2013). *Univa Grid Engine*. Available online at: http://www.univa.com/products/grid-engine.php (Visited 5.11.2013).

Vetterli, M., and Kovačević, J. (1995). *Wavelets and Subband Coding*, Vol. 87. New Jersey: Prentice Hall PTR Englewood Cliffs.

Vinck, M., Oostenveld, R., van Wingerden, M., Battaglia, F., and Pennartz, C. (2011). An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias. *Neuroimage* 55, 1548–1565. doi: 10.1016/j.neuroimage.2011.01.055

Vinck, M., van Wingerden, M., Womelsdorf, T., Fries, P., and Pennartz, C. (2010). The pairwise phase consistency: a bias-free measure of rhythmic neuronal synchronization. *Neuroimage* 51, 112–122. doi: 10.1016/j.neuroimage.2010.01.073

Wilson, S. M., Molnar-Szakacs, I., and Iacoboni, M. (2008). Beyond superior temporal cortex: intersubject correlations in narrative speech comprehension. *Cereb. Cortex* 18, 230–242. doi: 10.1093/cercor/bhm049

Worsley, K. (2008). *SurfStat. A Matlab Toolbox for the Statistical and Multivariate Surface and Volumetric Data Using Linear Mixed Effects Models and Random Field Theory*. Available online at: http://www.math.mcgill.ca/keith/surfstat/ (Visited 5.11.2013).

Yoo, A., Jette, M., and Grondona, M. (2003). "SLURM: simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science*, Vol. 2862, eds D., Feitelson, L., Rudolph, and U., Schwiegelshohn (Berlin; Heidelberg: Springer), 44–60.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 07 November 2013; accepted: 09 January 2014; published online: 31 January 2014.

Citation: Kauppi J-P, Pajula J and Tohka J (2014) A versatile software package for inter-subject correlation based analyses of fMRI. Front. Neuroinform. ***8****:2. doi: 10.3389/fninf.2014.00002*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 12 September 2014
doi: 10.3389/fninf.2014.00074

Cyberinfrastructure for the digital brain: spatial standards for integrating rodent brain atlases

Ilya Zaslavsky[1]*, Richard A. Baldock[2] and Jyl Boline[3]

[1] *San Diego Supercomputer Center, University of California San Diego, La Jolla, CA, USA*
[2] *MRC Human Genetics Unit, Institute of Genetics and Molecular Medicine, University of Edinburgh, Edinburgh, UK*
[3] *Informed Minds, Wilton Manors, FL, USA*

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Allan MacKenzie-Graham, University of California Los Angeles, USA
Christiaan P. J. De Kock, VU University Amsterdam, Netherlands

****Correspondence:***
Ilya Zaslavsky, San Diego Supercomputer Center, University of California San Diego, 9500 Gilman Dr., La Jolla, CA 92093-0505, USA
e-mail: zaslavsk@sdsc.edu

Biomedical research entails capture and analysis of massive data volumes and new discoveries arise from data-integration and mining. This is only possible if data can be mapped onto a common framework such as the genome for genomic data. In neuroscience, the framework is intrinsically spatial and based on a number of paper atlases. This cannot meet today's data-intensive analysis and integration challenges. A scalable and extensible software infrastructure that is standards based but open for novel data and resources, is required for integrating information such as signal distributions, gene-expression, neuronal connectivity, electrophysiology, anatomy, and developmental processes. Therefore, the International Neuroinformatics Coordinating Facility (INCF) initiated the development of a spatial framework for neuroscience data integration with an associated Digital Atlasing Infrastructure (DAI). A prototype implementation of this infrastructure for the rodent brain is reported here. The infrastructure is based on a collection of reference spaces to which data is mapped at the required resolution, such as the Waxholm Space (WHS), a 3D reconstruction of the brain generated using high-resolution, multi-channel microMRI. The core standards of the digital atlasing service-oriented infrastructure include Waxholm Markup Language (WaxML): XML schema expressing a uniform information model for key elements such as coordinate systems, transformations, points of interest (POI)s, labels, and annotations; and Atlas Web Services: interfaces for querying and updating atlas data. The services return WaxML-encoded documents with information about capabilities, spatial reference systems (SRSs) and structures, and execute coordinate transformations and POI-based requests. Key elements of INCF-DAI cyberinfrastructure have been prototyped for both mouse and rat brain atlas sources, including the Allen Mouse Brain Atlas, UCSD Cell-Centered Database, and Edinburgh Mouse Atlas Project.

Keywords: digital atlases, atlas infrastructure, spatial data integration, brain coordinate systems, Waxholm space, atlas services, coordinate transformations

INTRODUCTION

Frequently asked questions in neuroscience are "where" in the brain something is happening, "what" is happening "here," and "what" is this structure. The extended version asks for similarity and association between biological processes and structures to understand complex observations. Most researchers, in one way or another, access information from a reference brain atlas and apply the associated material to their own datasets. This allows them to compare and analyze data within their own laboratories as well as in relation to outside sources. Mouse brain atlases were initially developed as paper atlases (Hof et al., 2000; Paxinos, 2004; Paxinos et al., 2007; Paxinos and Watson, 2009), and have been used in this form for many years to support spatial referencing in electrophysiology and other studies. Recently, atlas providers have put significant effort into organizing atlas information in digital form, creating digital brain atlases as collections of spatially and semantically consistent 2D images or 3D volumes with anatomical structure delineations and additional annotations. These atlases have been made accessible via desktop [e.g., MRM NeAT (http://brainatlas.mbi.ufl.edu/), Mouse Atlas Project (http://map.loni.usc.edu/), CIVM (http://www.civm.duhs.duke.edu/)] and online interfaces such as the Allen Brain Atlas (http://www.brain-map.org/), EMAP, (http://www.emouseatlas.org/emap/home.html), MBL (http://www.mbl.org/mbl_main/atlas.html)

Abbreviations: ABA, Allen Brain Atlas; AGEA, Anatomic Gene Expression Atlas; API, Application Programming Interface; CSW, Catalog Services for the Web; DAI, Digital Atlasing Infrastructure; EMAGE, Edinburgh Mouse Atlas Gene Expression database; GML, Geography Markup Language; INCF, International Neuroinformatics Coordinating Facility; MBAT, Mouse BIRN Atlasing Toolkit; OGC, Open Geospatial Consortium; POI, Point of Interest; SOA, Service-Oriented Architecture; SRS, Spatial Reference System; WHS, Waxholm Space; WaxML, Waxholm Markup Language; WIB, Web Image Browser; WPS, Web Processing Service.

Mouse Brain Atlas http://www.hms.harvard.edu/research/brain/atlas.html, Genepaint (http://genepaint.org/Frameset.html), Australian Mouse Brain Mapping Consortium (http://www.tissuestack.org), Rodent Brain WorkBench (http://www.rbwb.org/), Laboratory of Brain Anatomical MRI (http://lbam.med.jhmi.edu/), Knife-Edge Scanning Microscope Brain Atlas (http://kesm.cs.tamu.edu/), and SumsDB (http://sumsdb.wustl.edu/).

While such atlases have been internally consistent, they have been developed largely independently of one another. Without uniform conventions for brain data representation and access, users have limited ability to quickly answer questions such as "which atlas-based resources have images for a specified part of the brain," "what genes are expressed in a given tissue in atlases A and B, at a specified expression level," "compare spatial patterns of protein distribution across atlases C and D," or "what proteins are expressed in the projection domains of hippocampal neurons." Yet answering such questions becomes increasingly important in neuroscience and other domains as scientists attempt to integrate information and knowledge encapsulated in multiple information sources to test hypotheses or to infer novel associations and patterns in an atlasing environment (Bjaalie, 2002; Toga, 2002; Baldock et al., 2003; MacKenzie-Graham et al., 2003; Martone et al., 2004; Zaslavsky et al., 2004; Boline et al., 2008; Hawrylycz et al., 2011; Zakiewicz et al., 2011).

While this type of environment has been desired by many members of the neuroscience community for quite some time now, a spatial framework that enables interoperability between existing atlasing efforts and allows the addition of other spatially-tied data has not been built for technical, social, and financial reasons. Creating such an environment has been one of the foremost goals of the Digital Atlasing Program of the International Neuroinformatics Coordination Facility, INCF (Hawrylycz et al., 2009, 2011). Under this program, INCF has brought together a group of neuroscientists and technology experts to organize atlas resources, explore and outline best practices and recommendations, and design and guide the development of standards, information infrastructure, and tools for integrating digital brain atlases.

Use cases established over recent years[1] show that most neuroscientists want to have the ability to bring together and compare different types of information: explore a reference atlas, juxtapose it with their own data, and finally, link and compare their data to other datasets. For instance, researchers using immunohistochemistry to examine images for a specific protein may not have much anatomical information in the images. Applying atlas delineations from a canonical atlas to their images would let them examine and quantify the level of labeling in different brain areas. With this information, they may wish to run a quantitative analysis that compares their data to another resource, such as the Allen Brain Atlas and then visualize it in 3D.

The compendium of use cases allowed us to identify three groups of researchers based on their use of atlases (**Figure 1**). The most basic need is simply to find and examine information about their area of interest (**Figure 1**, User 1). Another group wants capabilities that include relating user resources with external canonical atlases based on spatial properties, such as location, shape or observed spatial pattern (**Figure 1**, User 2). Finally a number of users want to share their data with others such that image collections, 3D reconstructions, gene expression or other information they collected can be accessed online and used as a reference in a given spatial framework (**Figure 1**, User 3). While simply posting data online is possible, placing the information into a known spatial framework provides the ability to run novel analyses (Carson et al., 2005; Kovacević et al., 2005; Christiansen et al., 2006; Leergaard and Bjaalie, 2007; Lein et al., 2007; Ma et al., 2008; Aggarwal et al., 2009; Ng et al., 2009; Chuang et al., 2011) and to integrate data from different atlas-based resources (Baldock et al., 2003; MacKenzie-Graham et al., 2004; Martone et al., 2004; Boline et al., 2008; Lee et al., 2010; Hawrylycz et al., 2011). Most users want to do this at some point, but many have no idea how to even start the process. This is an extremely daunting task, due, to a large degree, to the complete lack or complexity of sharing conventions for atlas data and supporting data publication tools. Meeting the needs of all these users through the creation of a flexible, expandable, and accessible spatial framework for sharing atlas data has been one of the main goals of the INCF Digital Atlasing Program.

A key component of this open framework is a common publicly accessible 3D reference space, providing standard coordinates and serving as a spatial anchor for other existing rodent brain atlas resources (Hawrylycz et al., 2011). Such a canonical Waxholm Space (WHS) has been developed for C57BL/6J mouse (Johnson et al., 2010). In addition, two recent versions of WHS for the rat, one Sprague Dawley (Johnson et al., 2012) and one Wistar (Papp et al., 2013) have been created. The goal is to embed them as the rat spatial anchors of our framework, register them to each other and to create a mapping from mouse to rat. In addition to standardizing reference spaces, agreements about how location information is represented and exchanged between atlases must be established—these agreements are the foundation of software infrastructure that support publication, discovery, access, and integration of distributed atlas information.

We have developed the underlying principles and implemented a prototype of an open standards-based spatial data integration framework, the Digital Atlasing Infrastructure (DAI). This includes the backbone of the infrastructure itself, along with a few online applications and tutorials to enable neuroscientists to use and add to the infrastructure. We expect that a rich set of supporting tools will be developed over time by members of the neuroscience and neuroinformatics communities leveraging standards-based information exchange protocols tested in the prototype.

This article describes the DAI, including its rationale, components, and the current state of the system. We focus on the formal definition of coordinate systems and coordinate transformations for rodent brain, a service interface for DAI services, and a standards-based XML schema for encoding atlas information, called Waxholm[2] Markup Language (WaxML). It is followed by

[1] http://wiki.incf.org/mediawiki/index.php/Use_Case

[2] Named after Waxholm, a town in Sweden where the first meeting of the INCF atlasing task force was held in 2007.

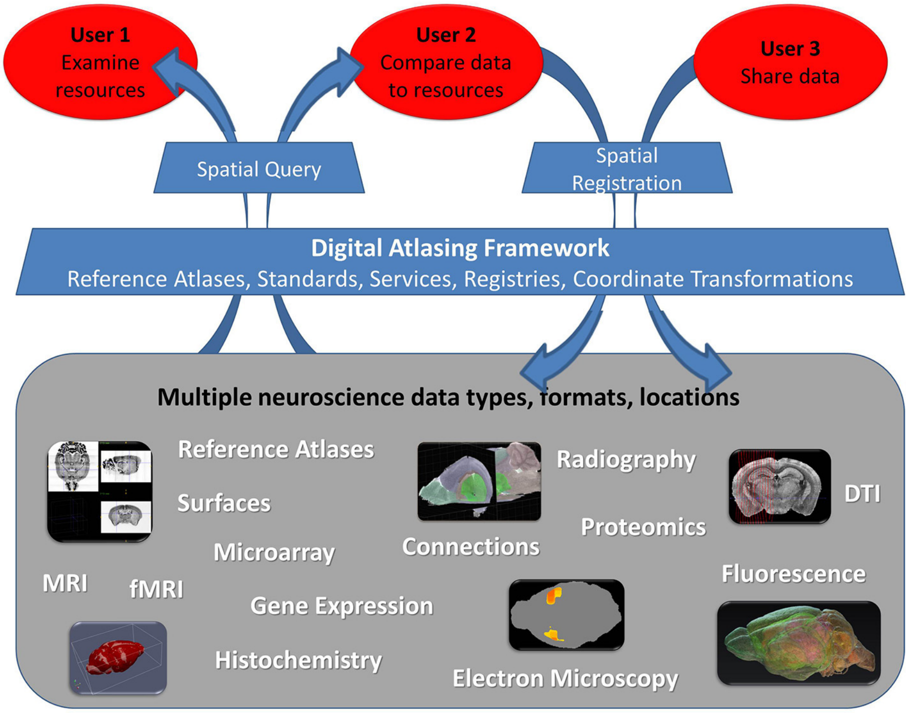


FIGURE 1 | Three user groups interacting with neuroscience data within the digital atlas framework. The framework should allow integration of datasets of various type, format, and location through the Digital Atlasing Infrastructure (DAI). Users are able to interact with this environment using DAI tools, which enable spatial query of data shared through this framework or addition of new data via spatial registration. Note that we differentiate data sharing mechanisms for User 2 and User 3: User 2 typically has a limited number of images and needs to register them primarily to explore other atlas sources spatially, while User 3 typically shares large volumes of spatially-referenced data within their group or to others, for the purpose of making it available for query and more automated analyses in a spatial framework. User 3 may even have their own reference atlas. The framework can be expanded to accommodate additional data types beyond those shown.

implementation details, and a description of a spatial registration pipeline, which illustrates how to extend the system with additional spatially-referenced data. Finally, we address the benefits of leveraging existing spatial integration frameworks and standards for atlas data integration, and future work.

DIGITAL ATLASING INFRASTRUCTURE: HIGH-LEVEL REQUIREMENTS AND MAIN COMPONENTS

The vision of brain atlases as interconnected gateways to large distributed and diverse atlas resources, including images, volume data, segmentations, gene expression, electrophysiology, behavioral, connectivity, other spatially-organized data, implies a number of design requirements:

- Atlases should be organized as spatial data sources, which support querying atlas data using spatial characteristics of their content, in particular by coordinates in a brain coordinate system.
- Information from multiple brain atlas sources should be available for searching and browsing, which typically involves indexing data elements in a spatial data registry.
- The spatial data and metadata must be accessible via standard protocols and in common formats, following established standard application programming interfaces (APIs) and information models. In addition, capabilities of each atlas resource should be advertised in a standard manner, so that different functions can be automatically invoked and chained to implement data integration and research workflows.
- DAI should incorporate transparent and easy to follow mechanisms for users to extend the system: by publishing and registering spatially-referenced atlas data, via standards-compliant spatial registration pipelines, and through annotation or segmentation.
- Brain atlas data must be accessible to a number of desktop and web-based data management, cataloging, analysis, visualization, and other applications that take advantage of the uniform APIs and information encodings. This model allows software developers the ability to use this resource for very different application needs.
- Ideally, most of the underlying services infrastructure will be invisible to the neuroscientists working through easy to use software tools that directly access DAI via standard APIs. As user needs evolve and the complexity of sharing or accessing data in a spatial framework increases, DAI will need continuing participation of neuroscience researchers to guide infrastructure development, through the INCF Digital Atlasing Program or similar mechanisms.

The DAI follows service-oriented architecture (SOA) principles (Erl, 2005; Josuttis, 2007), whereby atlas information becomes available via *atlas web services,* a collection of functions that deliver spatial and other information in standardized agreed-upon formats, thus alleviating the existing heterogeneity across different atlas resources. The high-level system architecture includes three key logical components (**Figure 2**):

(a) Atlas Hubs—an atlas data publication platform: a software stack for publishing neuroscience atlas data and web services, compliant with the WaxML schema and atlas services specification. An atlas hub may be maintained by an atlas-related project, or hosted by INCF as a proxy of a remote atlas resource.

(b) INCF Atlas Central—the central data discovery and integration platform: a catalog of atlas web services from multiple hubs, as well as other atlas-related data. Using standard catalog services, users and applications can search for appropriate web services across atlas hubs. In addition, the INCF Atlas Central system contains a special "central atlas hub" designed as a mediator for coordinate transformation services invoked across multiple hubs.

(c) Atlas Applications—the data synthesis and research platform: a collection of analysis, visualization, modeling, and other applications that consume standard atlas data and metadata (catalog) services, or are used to manage and update atlas information at a hub. Such applications include, for example, the INCF Scalable Brain Atlas and the UCSD Web Image Browser (WIB), developed by different DAI partners.

The initial focus of the atlasing infrastructure is limited to representation of anatomic features in the brain, brain reference systems and coordinate transformations, fiducial points and landmarks, and a few types of spatially referenced data and annotations that can be retrieved using point of interest (POI) requests. These functions fit the needs of our "User 1," those looking for spatially-linked data. In our review of existing online atlases of rodent brain we found significant heterogeneity in modalities, formats and functionality. Individual atlas resources may support different data types and use different metadata and data representations; they have been developed using different data collection methods; support different data retrieval, processing and other functions, and often adhere to different spatial and semantic frameworks. For example, a neuroscientist might want to use POI requests to find the name of the structure at this POI in WHS, the Allen Mouse Brain Atlas, or a Paxinos annotated atlas. They may wish to discover all available images in the vicinity of the POI regardless of atlases that contain them. However, some existing atlas resources may not support structure or image retrieval based on brain location; the structure names often belong to different vocabularies; and structure geometries depend on different delineation techniques, complicating cross-comparison. Similarly, any discovered images are likely to be in different formats and reflect different measurement modalities and instruments.

This heterogeneity presents an informatics challenge in developing an interoperable system for brain information that can work across multiple, independently managed, atlas information sources, processing services, and client applications. Hence, development of shared information models and data exchange protocols, and information brokers, is a central requirement for designing communication across DAI components. Establishing community consensus about information models and exchange protocols ensures that infrastructure components are structurally interoperable. Standards-compliance also enhances extensibility of the atlas infrastructure, by making it easier to incorporate standards-based software modules created by developers outside the DAI project. Consequently, maintenance of standards-based systems is usually less expensive, and expertise is easier to find because it does not have to come from a single group. In the long run, such systems evolve more easily with changes in technology, and are more economical

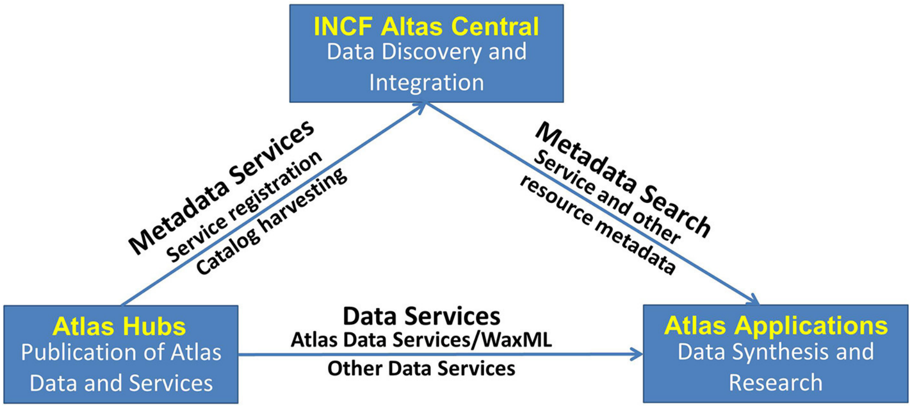

INCF Altas Central
Data Discovery and Integration
Metadata Services
Service registration
Catalog harvesting
Metadata Search
Service and other resource metadata
Atlas Hubs
Publication of Atlas Data and Services
Data Services
Atlas Data Services/WaxML
Other Data Services
Atlas Applications
Data Synthesis and Research

FIGURE 2 | High-level design of the INCF Digital Atlasing Infrastructure. The design follows the standard SOA "publish-find-bind" pattern, bringing together providers of atlas data and services, catalog and discovery services, and data synthesis and research applications. *Atlas Hubs* share their data via DAI-compatible services. *INCF Atlas Central* contains a catalog of what is available from the Atlas Hubs and also acts as a "translator" between the different spatial coordinates offered by the Atlas Hubs. Various *Applications* can be developed that use INCF Atlas Central to find what is available and then access the services offered by the Atlas Hubs. This SOA-based design allows significant flexibility in tool development.

as they encourage cooperation, competition, and prevent a software vendor lock-in (David and Greenstein, 1990; West, 2007).

Development of consensus about data sharing formats and protocols, and their community adoption, is a long process; therefore, one of the key requirements of the DAI is enabling evolution of the system to such standard conventions rather than enforcing rigid standards compliance from the start. As described in the next section, this approach is adopted in the choices for specifying and implementing atlas services, markup, and in defining spatial reference systems (SRSs) and transformations.

STANDARDIZATION OF SPATIAL REPRESENTATION AND SPATIAL DATA ACCESS TO RODENT BRAIN DATA

Three standard components need to be specified in an interoperable atlas infrastructure design: (1) a common spatial framework, (2) the structure of key information elements to be exchanged across atlases, and (3) the respective exchange protocols.

COMMON SPATIAL FRAMEWORK

Established paper atlases of rodent brain (Paxinos and Watson, 1998; Swanson, 1998; Hof et al., 2000; Paxinos and Franklin, 2001) include coordinate systems used to describe anatomic feature locations and relationships in terms of distance to key brain landmarks (e.g., bregma, midline) and neuroscience anatomical axes: dorsal-ventral, anterior-posterior, left-right. In some cases, such feature-based coordinate systems are combined with image-based coordinates, but most typically, for a collection of images forming an atlas, locations are only referenced by a slice index and by image coordinates within the slice. Due to a wide variety of imaging and processing techniques, and different physical properties of the sectioned brains, there is little consistency across such spatial descriptions, which makes it difficult to translate location information from one atlas to another and subsequently integrate data based on location in the brain except in the most cursory manner.

A similar problem has been recognized and resolved in geodesy, where many coordinate systems have been developed over the centuries for different purposes, at different resolutions, using different models of the earth, and allowing for different types of distortions (in direction, area, shape, distance). The solution involved several components:

(a) development of more accurate mathematical descriptions of the shape of the earth;
(b) creating precise and consistent models of projections as transformations from earth coordinates into various 2D and 3D digital representations;
(c) standardization of coordinate transformation descriptions (e.g., the OpenGIS Coordinate Transformation Service Implementation Specification, see http://www.opengeospatial.org/standards/ct);
(d) cataloging the available coordinate systems (e.g., the EPSG Geodetic Parameter Dataset); and
(e) development of widely used coordinate transformation packages (e.g., the General Cartographic Transformation Package).

Registries of coordinate systems and coordinate transformation libraries are foundational components of global spatial data infrastructure; they are accessed from multiple spatial information system software packages. For example, the geospatial SRS registry (http://www.epsg-registry.org/) contains definitions of thousands of SRSs. For each system, the description includes a code (e.g., EPSG:4326), which is used by process libraries, web services and other software applications to reference the SRS; name (e.g., World Geodetic System 1984 or WGS84), type of SRS (e.g., "geographic 2D"), specification of the "Area of Use" (e.g., "world"), as well as description of the underlying geodetic datum, projection conversion, and versions/revisions.

While definitions of brain coordinate systems differ significantly from geodetic coordinate systems, INCF DAI design borrows several key ideas from geospatial data infrastructure. As in geodesy, DAI recognizes a number of coordinate systems in different atlases, and does not mandate a single reference space. At the same time, WHS, being a publicly available open reference space, serves as a common and convenient "go-between" system much like latitude and longitude coordinates in a well-defined SRS (e.g., WGS84) are often used to transform coordinates between any two arbitrary systems. This allows us to use space rather than structural naming conventions to convey location. Structure names then become a type of information, which may be available at a location in the space of the brain, and may be different across atlases. For example, the same point location may be labeled as "Striatum dorsal region" in the Allen Mouse Brain Atlas, "Caudate putamen striatum" in the Paxinos atlas, or "Striatum" in WHS (**Figure 9B**), with names generally depending on image modality, delineation techniques, classification model, or adopted level of generality.

To create spatial infrastructure for brain atlases, we:

- developed a generic representation of a rodent brain coordinate space,
- compiled a registry of such coordinate systems,
- computed transformations between several existing reference spaces and implemented them as a set of standard services, and
- composed and implemented a workflow for deriving new coordinate systems and associated transformations between the new coordinate system and an existing one.

Table 1 lists several of the coordinate systems for rodent brain initially defined by the project and included in the SRS registry. These came from members of the atlasing community that were able to fairly quickly share their data within a spatial framework (e.g., User 3). **Figure 3** illustrates some of them, along with origin and axis orientation shown on each diagram with respect to neuroscience orientations, as well as units and spatial extent on each coordinate axis. Note the wide variability in coordinate systems used in the various atlases.

In the current DAI model, SRS descriptions are designed to provide sufficient information for neuroscientists to understand how the SRS is constructed with respect to neuroscience orientation and key anatomic features, and evaluate its applicability as an alignment target. Therefore, SRS descriptions include:

Table 1 | Spatial reference system core characteristics for the mouse atlases currently registered in DAI.

Code	Name	SRS family	Version	Species	SRS description
INCF:0001	Mouse_WHS_0.9	WHS	0.9	Mouse	WHS initial version, with origin in the back-left-bottom corner
INCF:0002	Mouse_WHS_1.0	WHS	1.0	Mouse	WHS with origin shifted to the intersection of midline and the center of anterior commissure
INCF:0100	Mouse_ABAvoxel_1.0	ABAvoxel	1.0	Mouse	SRS used in the Allen Mouse Brain Atlas 3D model (circa 2005)
INCF:0101	Mouse_ABAreference_1.0	ABAreference	1.0	Mouse	SRS in the Allen Mouse Brain Atlas reference atlas
INCF:0102	Mouse_AGEA_1.0	AGEA	1.0	Mouse	SRS used in the Allen Mouse Brain Atlas gene expression module, a derivative of ABAvoxel
INCF:0200	Mouse_Paxinos_1.0	Paxinos	1.0	Mouse	SRS in the Paxinos and Franklin (2001) stereotaxic atlas of the mouse brain
INCF:0300	Mouse_EMAP-T23_1.0	EMAP-T23	1.0	Mouse	A T23 model of EMAP developing mouse atlas

- coordinate system origin,
- coordinate axes and measurement units,
- pointer to the SRS's reference implementation,
- specification of the region of validity and valid extents along each of the coordinate axes,
- the author of the SRS, and
- how the SRS was derived from another coordinate system, if applicable.

The "Region of Validity" is a characteristic analogous to the "Area of Use" in the EPSG registry. In addition to the whole brain coordinate systems registered so far, DAI allows users to register additional SRS defined more precisely for smaller regions in the brain, using the workflow described later in the paper. For such SRS, the region of validity is defined by an anatomic structure or a group of structures, and valid spatial extents along the X, Y, and Z axes. The DAI ability to manage multiple coordinate systems, both for the whole brain and local to an anatomic structure, facilitates spatial integration of neuroimaging information across different modalities and resolution levels, as DAI users can select an appropriate reference space (e.g., with matching resolution, region of validity, and modality) to explore available data or to register their own data.

The coordinate system registry contains an additional mandatory table called "Orientation," which provides interpretation of neuroscience coordinate axes or their derivatives used to define X, Y, and Z coordinates in the SRS table. These axes may be simple (e.g., describing straight dorsal-ventral, anterior-posterior, or left-right orientations), or complex. The latter could be used to describe orientations in the developing brain (where the posterior and anterior orientations may be described as curves rather than straight lines) or volumes/images that are tilted or otherwise transformed with respect to canonical anatomical terms of location. Note that such a description should be sufficient for neuroscientists to understand how the coordinate system was constructed, and roughly orient it with respect to other SRS, but in most cases will be insufficient for deriving coordinate transformations: the latter are computed and registered separately.

Additional tables in the SRS registry are optional and include: "Structure," "Fiducial," and "Slice." "Structure" includes descriptions of anatomic structures delineated in 2D or 3D, along with references to structure vocabulary and a spatial object describing the structure, or a method for deriving the latter. "Fiducial"s are recognizable points or higher-dimensional features generally derived from anatomic structures or their relationships, which can be used to automatically relate one SRS to another, or recommend point pairs for fine alignment. Finally, "Slice" is used when the SRS is defined through a collection of 2D plates with segmented structures rather than by a 3D volume; it contains descriptions of individual slices, or plates, that together form the 3D atlas. A more complete description of tables in the SRS registry can be found at http://wiki.incf.org/mediawiki/index.php/SRS_Registry.

In INCF-DAI, information from this registry (encoded in WaxML) is currently available via several atlas service requests that are supported by all atlas hubs (*ListSRSs* and *DescribeSRS*). WaxML and the atlas services are described in subsequent sections of the paper.

In addition to the registry of SRSs, INCF-DAI also maintains a registry of coordinate transformations between known coordinate systems. While there is no requirement for a specific coordinate system to be implemented by all atlas sources, there is a requirement that any new user-supplied atlas data are registered to at least one known coordinate system. For practical reasons, within INCF-DAI it is recommended that at least forward and inverse transformations between all SRSs and WHS are supported, since, with WHS as an intermediary, coordinate transformation between any two SRSs that do not have direct mapping, would require two steps. While this is not a strict requirement within DAI, limiting the number of steps in a composite transformation reduces any mapping errors that might occur due to registration.

Different procedures, depending on the representation (collection of 2D slices, 3D model) and known relationships between reference spaces, have been used to derive forward and inverse transformations between pairs of registered coordinate systems. Registration methods include those implemented in ITK/ANTS (Avants et al., 2011) (http://www.picsl.upenn.edu/ANTS) for 3D volume registration, warping of individual 2D slices to matching slices in a 3D volume using thin plate spline calculations, and piecewise linear mapping functions for selected 3D atlas slices to a 2D plate. In the absence of good assessment techniques for transformation accuracy between two images

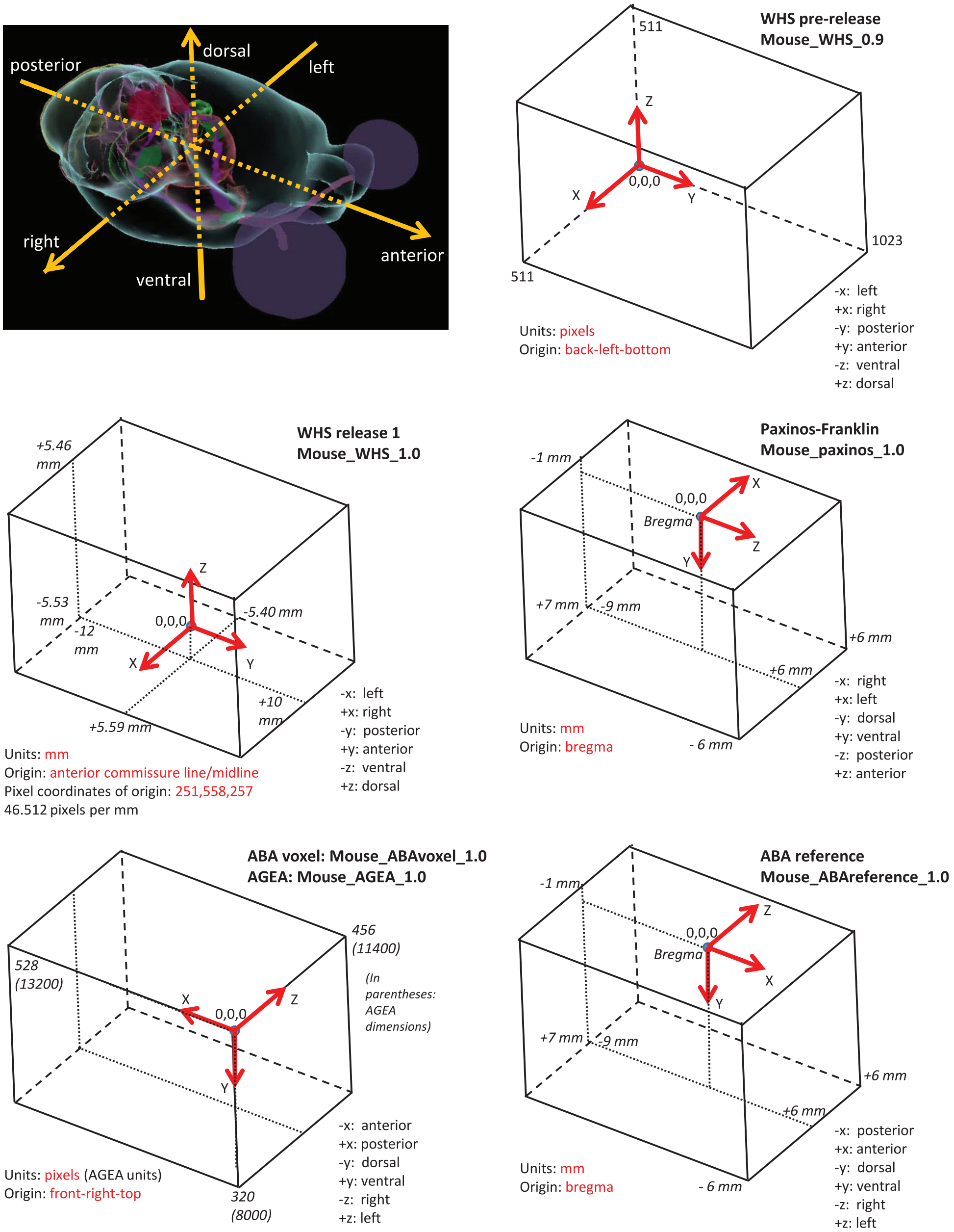


FIGURE 3 | Selected coordinate systems for mouse brain of several common atlas reference spaces. All coordinate systems (boxes) are shown relative to the anatomical picture of the mouse brain shown in the upper left corner. Note the variability in direction and origin of the atlases. Much of the variability arose from practical reasons (e.g., stereotaxic surgery) or because of the data collection method used.

(besides visual inspection of resultant alignment), inverse transformation consistency is computed for each translation function and returned to the user as part of coordinate transformation responses (*TransformPOI*). Using the spatial alignment workflow provided within DAI, or any other similar workflow, users are encouraged to develop new transformations or additional versions of existing transformations to improve registration and coordinate transformation accuracy for their region of interest, make them available via atlas services, and register them in the registry of transformations.

WAXHOLM MARKUP LANGUAGE

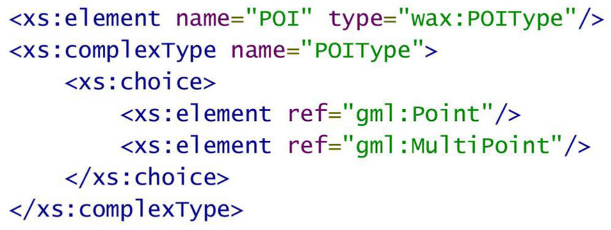
Existing atlases often present examples of different implementations of closely related functionality, or multiple ways of encoding similar types of data. For example, gene expression information might be labeled as "high," "low," or "none" within a neural structure or quantified as a number in a structure or region of space. An example is the information available from Allen Brain Atlas's AGEA (Anatomic Gene-Expression Atlas) via its *GeneFinder* requests, which return numeric normalized expression value at a location in space (see http://help.brain-map.

org/download/attachments/2818169/InformaticsDataProcessing.pdf?version=1&modificationDate=1319667590884, p. 5–6). In contrast, the Embryonic Edinburgh Map Atlas project (EMAP) framework holds EMAGE data, where expression levels are returned with keywords for a selected region such as "strong," "detected," or "not detected" (Baldock et al., 2003; Christiansen et al., 2006). This is likely the more common way of representing this type of information, but even these designations may be assigned using various methods. At the same time, there have been several efforts to develop gene expression markup, including MAGE-ML (Spellman et al., 2002) (http://www.mged.org/Workgroups/MAGE/mage.html), and MINiML (Barrett et al., 2007) (http://www.ncbi.nlm.nih.gov/geo/info/MINiML.html). This illustrates some of the diversity of perspectives, research approaches and methods of neuroscientists. Conveying information about both the methods and results in a formal schema that is human and machine readable and also acceptable to different atlas publishers is highly desirable, but extremely difficult. As discussed above, our strategy to overcome this hurdle is to develop an information system that supports convergence to a consensus representation rather than mandates a single representation from the start. While allowing atlas hub providers a degree of freedom, this approach recommends standard structures and semantics appropriate for exchange of spatial information in the brain and also allows continual updating and improving of representations as methods and analyses evolve.

WaxML is the information model used to express key elements from atlas hubs. It offers formal semantics for atlas information, defining valid elements, their attributes and relationships. Specifically, it provides type definitions for basic atlas classes that describe SRSs, spatial transformations and key geometry types (**Table 2**). It also gives output schemas for brain location-based service requests, which include structures for anatomic features, gene expression, images and image collections, annotations, and other objects returned in response to POI-based requests. As mentioned above, we allow for differently structured responses to similar requests, due to specific implementations and approaches adopted by different atlases, as long as geometric representations remain consistent and interoperable.

WaxML borrows spatial object descriptions from the Open Geospatial Consortium (OGC) Geography Markup Language, GML (Portele, 2007), an international standard for spatial data encoding (ISO 19136). In particular, representation of spatial features and locations in the brain follows the GML simple features profile (Van den Brink et al., 2012). For example, a GML Point construct is used to encode points of interest (POI) (**Figure 4**), following POI definition in WaxML schema (in WaxML_Base.xsd), which references GML representation of points and multipoints—the latter construct is used when the request is to process an array of points rather than a single point of interest (**Figure 5**).

As an application schema of GML, the WaxML schema is compiled with GML 3.2.1, which is available at http://schemas.opengis.net/gml. Leveraging proven and well-documented standard geometric descriptions allows WaxML developers to reuse a range of common open source software libraries, and create

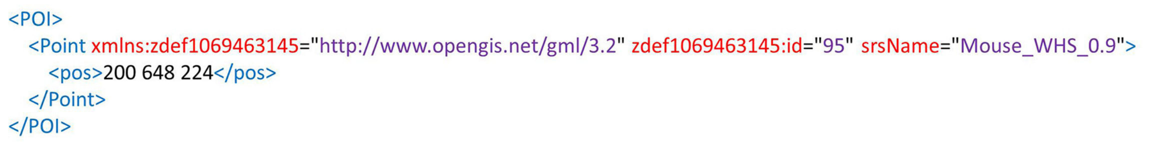


```
<xs:element name="POI" type="wax:POIType"/>
<xs:complexType name="POIType">
    <xs:choice>
        <xs:element ref="gml:Point"/>
        <xs:element ref="gml:MultiPoint"/>
    </xs:choice>
</xs:complexType>
```

FIGURE 5 | Fragment of WaxML_Base.xsd schema referencing GML Point and MultiPoint constructs.

Table 2 | Common WaxML schema components (see https://code.google.com/p/incf-dai/).

Schema name	Description
CoordinateTransformationCommon	Constructs related to coordinate transformation information, including transformation code, implementing atlas hub, input SRS, output SRS, transformation performance, order of transformations in a transformation chain
SrsCommon	Constructs related to spatial reference systems (SRS), as described in Section Common Spatial Framework
WaxML_Base	Basic constructs used across WaxML, specifying base input and response types, geometry types, and key enumerations



```
<POI>
  <Point xmlns:zdef1069463145="http://www.opengis.net/gml/3.2" zdef1069463145:id="95" srsName="Mouse_WHS_0.9">
    <pos>200 648 224</pos>
  </Point>
</POI>
```

FIGURE 4 | Representation of point of interest (POI) using the GML Point construct. Note that spatial reference system name is a mandatory attribute of Point.

software interoperable with multiple existing client and server codes, while focusing on classes that are specific to brain atlases.

ATLAS SERVICES

The atlas service interface specification is another key standard that forms the backbone of INCF-DAI. Atlas services are web functions that support querying and updating brain atlas resources offered by an atlas hub, returning information in WaxML-encoded documents.

The atlas services follow OGC Web Processing Service (WPS) interface standard (http://www.opengeospatial.org/standards/wps), which provides a framework for describing, invoking and chaining web requests, specifically oriented to spatial data processing functions. The key advantage of WPS for atlas services at this stage is that the services are self-describing (via the mandatory *GetCapabilities* and *DescribeProcess* requests), and the descriptions include information about the inputs and the output schema. The set of service requests may vary between atlas hubs, reflecting differences in implementation of atlas resources. Adherence to the WPS standard establishes initial structural consistency across different atlas services, and lets application developers reuse multiple standard service libraries (including WPS authoring libraries in Java and Python), client applications, and service metadata registries.

The general format of a WPS request is:

```
http://<server-path>/<HostServiceController>?Service=
WPS&version=1.0.0
   &Request=<WPS_Request>
   &Identifier=<identifier_name>
   &ResponseForm={format}
   &DataInputs={Encoded Inputs}
```

where WPS_Request may be one of *GetCapabilities*, *DescribeProcess* or *Execute* statements; the <identifier_name> clause refers to the function (process) to be invoked, such as *Get2DImagesByPOI;* ResponseForm specifies the output format of the response; and DataInputs includes a list of input values.

The WPS standard, and standard libraries implementing WPS, offers a few additional capabilities useful for DAI, including the built-in ability to manage large volume processing on servers without returning processing results to the client application (via an optional &*storeExecuteResponse*=*true* clause), execute chains of functions, request status updates for long-running processes (via the optional &*status*=*true* clause), and return lineage information in service responses (via the optional &*lineage*=*true* clause).

A number of core and optional INCF-DAI atlas service requests have been defined, as described below (see http://wiki.incf.org/mediawiki/index.php/Atlas_Services for additional details).

Core atlas service requests

These atlas service requests include key operations enabling exchange of location information in DAI. They provide basic information about hub capabilities and supported functions as well as coordinate systems and transformations, and enable execution of transformations and transformation chains.

- Service capability descriptions: *GetCapabilities* and *DescribeProcess*. These requests, mandated by the WPS standard, provide a list of functions (processes) included in an atlas service, and their descriptions.
- Descriptions of SRSs hosted by an atlas service implemented at an atlas hub: *ListSRSs, DescribeSRS*. These requests return coordinate system origin, units, definitions of coordinate axes and other SRS metadata (see Common Spatial Framework) formatted as WaxML documents. The functions are implemented at all atlas hubs that publish data in a coordinate system unique to that hub.
- Spatial transformations: *ListTransformations, TransformPOI.* The first of these functions lists forward and inverse coordinate transformations implemented at a hub. Additional coordinate systems and transformations can be automatically added to the system as new images and volumes are registered using the registration workflow described in Section Data Publication: the Spatial Registration Workflow. The second function executes a specified transformation for given coordinates of a point of interest (POI) or an array of points, generating coordinates of the POI or a POI array in the target atlas space.
- A client application may request a coordinate transformation that involves several steps. For instance, translating coordinates between reference plates in the Paxinos mouse atlas in stereotaxic coordinates, and reference plates of the Allen Mouse Brain Atlas, requires a chain of transformations that involve WHS, AGEA, and Allen Mouse Brain Atlas voxel model as intermediary coordinate spaces. An optimal transformation path is generated by *GetTransformationChain* at the central atlas hub, as described in Section Implementation. This chain could be avoided if direct registrations existed between all of the reference atlases; however, this is not practical, so in many cases this direct mapping does not exist.
- Some atlas hubs may provide sparse content for certain types of data, hence atlas queries may return empty responses. For example, requesting annotations or 2D images available at a given POI may yield empty responses, especially in the early phases of DAI development. To optimize POI-based requests, general information about availability of different types of registered objects (images, annotations, gene expression data, etc.) in the vicinity of a given POI, across multiple atlas hubs, should be available. This information is returned on the *GetObjectsByPOI* request implemented at the central atlas hub, which returns a list of POI-based methods that would result in non-empty responses.

Optional atlas service requests

These atlas service requests are not mandatory but are likely to be implemented at one or several atlas hubs. Typically, these additional requests for individual hubs reflect information content provided by the atlas, and are implemented as WPS service wrappers over existing native functionality of the atlas resource.

These include such POI-based requests as *GetStructureNamesByPOI, Get2DImagesByPOI; GetCorrelationMapByPOI; GetGenesByPOI*, *GetAnnotationsByPOI*, which accept a point of interest in any known SRS and return a respective WaxML document from a given atlas service. For example, the

GetStructureNamesByPOI method supports structure lookup for a canonical set of segmentations defined for an atlas, returning WaxML descriptions of structures found in the vicinity of a POI, along with geometric properties of each structure if available. While at this stage DAI is primarily concerned with coordinate information exchange and spatial requests (e.g., POI-based requests), atlas hubs may also include queries that don't involve brain location, e.g., queries by structure name, gene name, or similar.

As discussed earlier, the ability to have different sets of functions published by different hubs is a design requirement of DAI, as the initial goal is to standardize treatment of coordinate systems and location information, and create a framework in which the community can converge, over time, toward a common set of POI-based functions, related semantic functions, and the structure of requests and returned schemas.

IMPLEMENTATION

As discussed earlier, a working prototype of INCF-DAI is implemented as a network of atlas hubs hosting atlas web services, the central metadata registry, which maintains a catalog of atlas resources, and a number of client applications that consume atlas service requests and use the results to integrate information from atlas hubs for analysis and visualization (**Figure 2**). These components are described below.

ATLAS HUBS

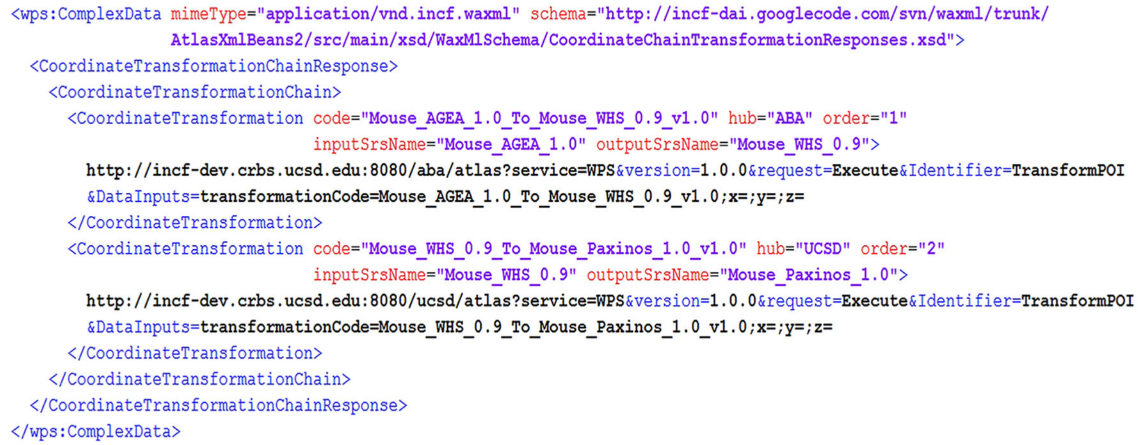
The atlas services have been implemented for five hubs: Allen Brain Atlas mouse hub, UCSD Cell-Centered Database hub, Edinburgh Mouse Atlas Project hub, a WHS mouse hub, and a central INCF atlas hub. In addition, rudimentary services with minimum set of functions have been setup for the two WHS rat hubs discussed earlier, though POI-based requests are not yet available for them. Any group that also wants to share their spatially-linked data in this manner may also consider setting up an atlas hub (User 3). As outlined in Section Core Atlas Service Requests, the hubs present service capability descriptions, SRSs unique to the hub, and coordinate transformations between these SRS and one or more globally-known coordinate system, such as WHS. The criterion is that for each hub publishing atlas data in a unique SRS, there should be at least one set of forward and inverse transformations that can be ultimately (i.e., via a sequence of transformations) connected with WHS, which in turn is maintained at the WHS hub. For example, the Allen Brain Atlas hub publishes three coordinate systems; the Allen Mouse Brain Atlas reference plates (ABAreference), Allen Mouse Brain Atlas 3D volume (ABAvoxel), and AGEA, in addition to several pairs (forward and inverse) of coordinate transformations: between ABAreference and ABAvoxel, between ABAvoxel and AGEA, and between ABAvoxel and WHS.

Besides these core functions, atlas hubs publish different sets of service methods, typically implemented as WPS wrappers over native atlas functions offered by their databases. For example, the ABA hub includes such functions as *Get2DImagesByPOI; GetCorrelationMapByPOI; GetGenesByPOI*, which wrap native ABA or AGEA functions (e.g., AGEA's GeneFinder interface takes coordinates of a seed point in AGEA coordinates as input).

In addition to hubs that publish specific atlas resources and/or coordinate systems and transformations, there is a special "central atlasing hub," which serves as a query mediator across other hubs and manages coordinate translations that involve more than one hub. It hosts a standard set of WPS-based atlas functions, which accept POI-based requests and translate them into respective web service requests against all registered hubs, then unions the responses before returning them to the user application. For example, a user may request a list of all 2D images available for a particular part of the brain from all atlas sources that support the *Get2DImagesByPOI* (illustrated in **Figure 9**). Information about all hubs that support this request is available because the atlas web service has been registered in the central service registry (see The INCF Central Metadata Registry and Discovery Portal for Atlas Resources), and lists of supported functions from each hub have been harvested into the central catalog. With this information available to the mediating hub, it rewrites the initial *Get2DImagesByPOI* query into respective requests that are valid for each atlas source.

An additional useful feature of DAI is that information for POI in the brain can be requested in any known coordinate system, since SRSName is a mandatory part of a POI definition. Coordinate translation to SRS understood by each hub are performed automatically, with the help of the *GetTransformationChain* request implemented at the mediator hub. This request uses information about all registered coordinate systems (which is harvested into the central database from all atlas services via *ListSRSs* calls) to construct an optimal sequence of coordinate translations from the POI included in user request, to target SRSs that a hub can process. The sequence of transformations is then executed as a series of TransformPOI calls. This processing is done behind the scenes, effectively allowing users and applications to issue service requests against any POI-based service in any known coordinate system. For example, a service request may use a POI in the coordinates of the Allen Mouse Brain AGEA, and expect it to be translated into the coordinate space of the (Paxinos and Franklin, 2001) mouse brain atlas, for querying atlas hubs that support the latter coordinate system. The respective *GetTransformationChain* request will generate a series of coordinate transformations such as the one shown in **Figure 6**, which involve a sequence of TransformPOI requests at the ABA and UCSD atlas hubs.

In the DAI prototype project, we used Deegree WPS libraries (http://www.deegree.org/) to develop and configure atlas services. This open source software implements OGC WPS 1.0.0, and configures standard WPS *GetCapabilities* and *DescribeProcess* requests based on a list of process providers, which represent containers for processes (functions) written in Java. The initial processes to publish through this mechanism include *ListSRSs* and *DescribeSRS* functions. Next, the hub author generates forward and inverse coordinate transformations that connect each of the new SRSs with WHS or another previously registered coordinate system, and makes this information available via *ListTransformations* and *TransformPOI* functions. After that, additional POI-based requests are implemented as appropriate for the types of resources to be published through the hub, using the same Java process containers. Other WPS development



```
<wps:ComplexData mimeType="application/vnd.incf.waxml" schema="http://incf-dai.googlecode.com/svn/waxml/trunk/
                 AtlasXmlBeans2/src/main/xsd/WaxMlSchema/CoordinateChainTransformationResponses.xsd">
  <CoordinateTransformationChainResponse>
    <CoordinateTransformationChain>
      <CoordinateTransformation code="Mouse_AGEA_1.0_To_Mouse_WHS_0.9_v1.0" hub="ABA" order="1"
                                inputSrsName="Mouse_AGEA_1.0" outputSrsName="Mouse_WHS_0.9">
        http://incf-dev.crbs.ucsd.edu:8080/aba/atlas?service=WPS&version=1.0.0&request=Execute&Identifier=TransformPOI
        &DataInputs=transformationCode=Mouse_AGEA_1.0_To_Mouse_WHS_0.9_v1.0;x=;y=;z=
      </CoordinateTransformation>
      <CoordinateTransformation code="Mouse_WHS_0.9_To_Mouse_Paxinos_1.0_v1.0" hub="UCSD" order="2"
                                inputSrsName="Mouse_WHS_0.9" outputSrsName="Mouse_Paxinos_1.0">
        http://incf-dev.crbs.ucsd.edu:8080/ucsd/atlas?service=WPS&version=1.0.0&request=Execute&Identifier=TransformPOI
        &DataInputs=transformationCode=Mouse_WHS_0.9_To_Mouse_Paxinos_1.0_v1.0;x=;y=;z=
      </CoordinateTransformation>
    </CoordinateTransformationChain>
  </CoordinateTransformationChainResponse>
</wps:ComplexData>
```

FIGURE 6 | A fragment of *GetTransformationChain* response. The response describes transformations from the Allen Mouse Brain AGEA (Mouse_AGEA_1.0) to the coordinate space developed in the (Paxinos and Franklin, 2001) mouse brain atlas (Mouse_Paxinos_1.0). It includes two *TransformPOI* request templates (with X, Y, Z coordinates left blank) served by two different atlas hubs: the ABA hub and the UCSD hub. The two *TransformPOI* service requests need to be made in sequence to execute the transformation chain. Note that the Mouse_WHS_0.9 coordinate space serves as the intermediate space for the two transformations: from AGEA to WHS 0.9 and then from WHS 0.9 to the target reference space of (Paxinos and Franklin, 2001).

libraries can be used as well, such as PyWPS (in Python, http://pywps.wald.intevation.org/) or ZooWPS (multiple languages, including C/C++, Fortran, Java, Python, PHP, Perl, JavaScript: http://www.zoo-project.org/).

THE INCF CENTRAL METADATA REGISTRY AND DISCOVERY PORTAL FOR ATLAS RESOURCES

INCF Atlas Central, hosting INCF-DAI portal and catalog, and a set of central registries (metadata, list of reference spaces and transformations) is the primary metadata registration, discovery, and integration platform. It is configured to periodically harvest information from individual atlas hubs via *GetCapabilities*, *DescribeProcess*, *ListSRSs*, and *ListTransformations* requests.

Atlas service metadata, as well as metadata for other types of registered resources (atlas-related image services, web-accessible folders with file collections, individual downloadable files, web sites, offline data, other standard catalog services, etc.), is organized in a central catalog, which is compliant with an international standard for spatially-enabled catalogs called *OGC Catalog Services for the Web* (CSW) (http://www.opengeospatial.org/standards/cat). This standard defines the request and response protocol for searching, adding, updating, and deleting catalog records. This CSW catalog is the core component of the INCF-DAI portal. The portal is implemented using open source Geoportal Server (http://sourceforge.net/projects/geoportal/) software, which is pre-configured to recognize standard service descriptions such as WPS, supports regular harvesting and updating registered resources of known types, and lets users browse and query atlas resource online.

We have customized the portal to support atlas-specific data types such as 2D images, segmentations, 3D volumes, connectivity data, and segmentations (**Figure 7**) and integrated it with several atlas client applications including WIB and Scalable Brain Atlas visualization clients. Because of the adoption of the CSW standard, the portal can be easily federated with other CSW-compliant portals, so that resources registered with one of the portals can be queried through another one.

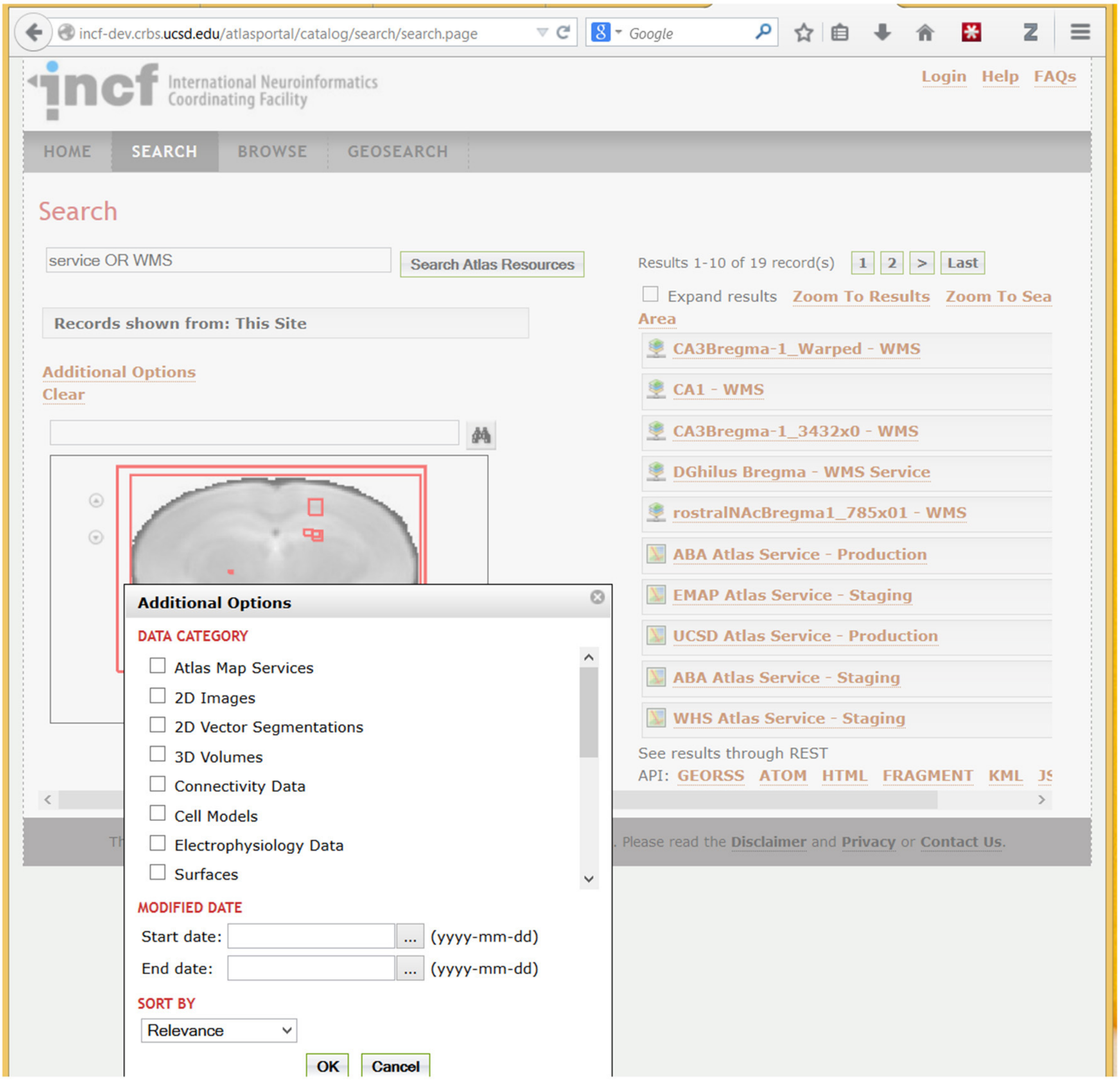
CLIENT APPLICATIONS ACCESSING ATLAS WEB SERVICES

Besides the atlas portal, resources registered in DAI can be accessed from a number of web applications (several shown in **Figure 8**). These applications make use of atlas service methods including coordinate translations and POI based requests. For example, WIB (Orloff et al., 2013) allows users to browse multiple atlas sections in three dimensions, and displays segmented anatomic features over high-resolution brain images (**Figure 9**). Users can zoom in to a POI and use it to query available atlas services and retrieve resources available from individual atlas hubs, or through the "central" atlas service, which spawns requests to all registered hubs and unions responses in a single output. The DAI coordinate translation services (*TransformPOI*) have also been used in the Scalable Brain Atlas (Bakker et al., 2010) (http://scalablebrainatlas.incf.org/), the Mouse BIRN Atlasing Toolkit (MBAT) (Ruffins et al., 2010) and the Whole Brain Catalog (Larson et al., 2010) (www.wholebraincatalog.org). In addition, a Python API accessing atlas web services has been developed (http://software.incf.org/software/incfdai?searchterm=python+DAI).

With these applications, users can compare anatomic feature descriptions, gene expression and other types of data available in different atlases and at different locations of interest. The Python wrapper also makes it easy for researchers to develop their own applications that take advantage of atlas services and the DAI framework.

DATA PUBLICATION: THE SPATIAL REGISTRATION WORKFLOW

The key DAI challenge is making the system extensible, to let users easily register and align their own data with existing atlases, add



incf-dev.crbs.ucsd.edu/atlasportal/catalog/search/search.page

service OR WMS

Results 1-10 of 19 record(s)

Records shown from: This Site

See results through REST API: GEORSS ATOM HTML FRAGMENT KML JS

Google
incf International Neuroinformatics Coordinating Facility
Login Help FAQs
HOME SEARCH BROWSE GEOSEARCH
Search
Search Atlas Resources
1 2 > Last
□ Expand results Zoom To Results Zoom To Search Area
CA3Bregma-1_Warped - WMS
CA1 - WMS
CA3Bregma-1_3432x0 - WMS
DGhilus Bregma - WMS Service
rostralNAcBregma1_785x01 - WMS
ABA Atlas Service - Production
EMAP Atlas Service - Staging
UCSD Atlas Service - Production
ABA Atlas Service - Staging
WHS Atlas Service - Staging
Additional Options
Clear
Additional Options
DATA CATEGORY
□ Atlas Map Services
□ 2D Images
□ 2D Vector Segmentations
□ 3D Volumes
□ Connectivity Data
□ Cell Models
□ Electrophysiology Data
□ Surfaces
MODIFIED DATE
Start date: ... (yyyy-mm-dd)
End date: ... (yyyy-mm-dd)
SORT BY
Relevance
OK Cancel
. Please read the Disclaimer and Privacy or Contact Us.

FIGURE 7 | A fragment of the DAI portal interface showing search results and types of searchable data. The example search for "Service OR WMS" (in Search Atlas Resources entry) returns metadata records that contain these terms. WMS refers to the OpenGIS Web Map Service standard (http://www.opengeospatial.org/standards/wms), which is used by the UCSD Cell Centered DataBase (UCSD Hub) to provide online access to large spatially-registered 2D images; thus all images stored using this method are returned in this search. Spatial extents of the found resources, in brain coordinates, are shown as red rectangles over a coronal slice. Users can optionally search for specific atlas data types (under "Data Category") illustrated in the pop-up box in the lower left corner. In addition to search, the portal supports metadata browsing (under the Browse tab) and search of resources based on geographic location of the lab that published a resource (under the GeoSearch tab).

coordinate systems and transformations, and contribute additional data to an atlas hub. This is usually done to expand analysis options and/or to allow direct comparison to other spatially-linked resources (User 2). Thus, the system would not be complete without a prototype registration workflow for aligning user-supplied 2D images and image collections to INCF-DAI reference spaces. While image alignment tools and pipelines have been developed (e.g., ITK/ANTS, LONI Pipeline, Amira, Slicer, NeuroMaps, MBAT, etc.), they often can be difficult to install, only accept 3D volumes, or the registration transformation is not stored along with the original datasets in an easily accessible and reusable manner.

Our goal was to develop a lightweight and intuitive online registration system for individual 2D images that uses a slice of a canonical atlas as the target. The system would be able to process images that are poorly aligned or have other artifacts preventing a straightforward 3D reconstruction; and would generate DAI SRS descriptions and transformations that are stored in association with the dataset, as the workflow outcome. This last step is essential to being able to reuse this information for analytic or query purposes.

This workflow can be accessed from the atlas portal, but requires an INCF account. The main workflow steps are shown in **Figure 10**. In the first step, a collection of segmented images is

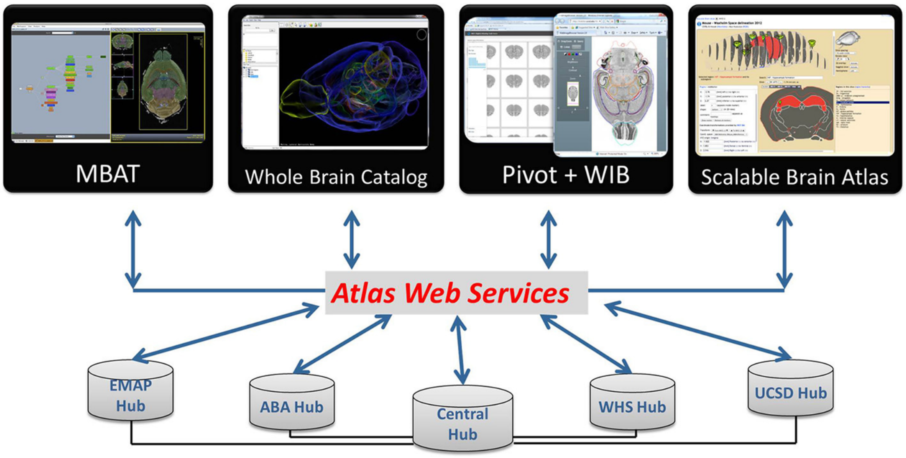


FIGURE 8 | DAI resources can be accessed via atlas web services from a number of atlas applications. Users can find what is available from INCF Central, and query atlas hubs via the Central Hub or directly through their web services. Online applications accessing atlas resources (the Whole Brain Catalog, PivotViewer, WIB, Scalable Brain Atlas) are available from the DAI portal.

uploaded into INCF DataSpace (http://www.incf.org/resources/data-space) via the INCF Atlas portal. The INCF DataSpace represents a common virtual storage space, where data from different INCF-affiliated labs are organized logically, abstracting specific storage resources used by each lab. It is implemented using iRODS (http://irods.org), which supports rule-based management of distributed files and file collections. In the context of INCF-DAI image registration workflow, iRODS rules are used to invoke initial processing of the uploaded images or image collections: generation of image pyramids, sub-sampled versions of the images, and image thumbnails. In addition, a *manifest* file is created, holding basic provenance information about the uploaded file collection and the processing steps.

Once the image files are packaged for processing, the content of the manifest file, and associated image thumbnails, are presented to the user in an image gallery page. From this page, users can visualize images in WIB or invoke the alignment interface. The latter component loads a sub-sampled version of the selected image into an alignment tool called *Jibber*. Jibber lets the user select a matching reference plate from a canonical atlas (in the current version, Allen Brain Atlas mouse reference plates or WHS sections), then adjust the image to match the target atlas plate as closely as possible. The affine transformation steps are followed by thin plate spline transformation based on user-defined links that connect correspondence point pairs or *tie-points* on the image and the target atlas plate. The generated transformation coefficients are passed to an engine called *Jetsam*, which generates a warped image and stores it in iRODS. The warping engine has been implemented on a computer cluster, to ensure fast warping of very large images. Based on these computations, a coordinate system description is generated, along with forward and inverse transformations between the user-submitted images and the canonical atlas used as the registration target.

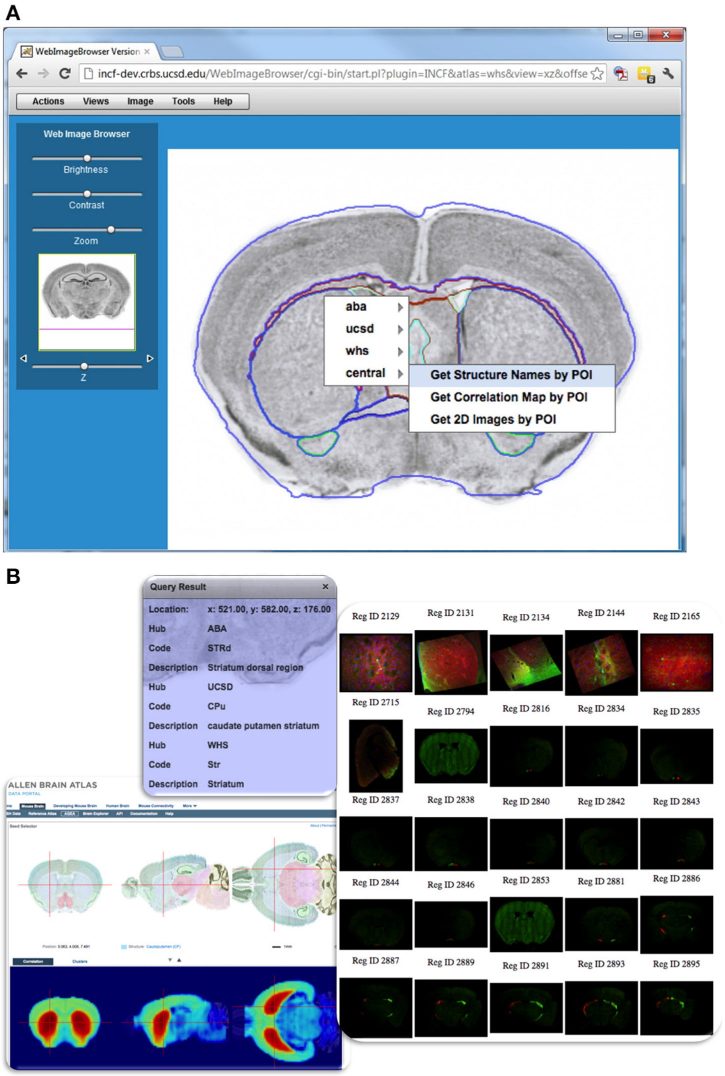
The SRS description and the transformations are updated as additional images from the image gallery are registered. This allows users to query other DAI information using spatial locations on their own images to retrieve structure names, discover available registered images, or explore gene expression and other data associated with user-defined POI, using an online tool such as WIB (**Figure 9**).

USING DAI

In addition to DAI technical components we have also developed tools and documentation to aid both neuroscientists and software developers interested in using or extending the system. Here we describe how these different users can find resources to access and contribute to the DAI.

The three types of neuroscientist users whose needs are addressed by DAI, are discussed in the introduction. User 1 wants to find and examine information about their area of interest, User 2 wants to compare their data to canonical atlases, and User 3 wants to contribute large datasets to a known spatial framework.

A simple query tool has been extended to fill the needs of User 1, WIB (see Section Client Applications Accessing Atlas Web Services); it can be found on the atlasing portal. The spatial registration workflow (Section Data Publication: the Spatial Registration Workflow) was created specifically to fit the needs of User 2. Finally, User 3 would need to first create an atlas hub, by setting up hub software, initially with a small set of mandatory atlas service functions, then defining additional spatial query functions appropriate for their data, and developing spatial transformations between hub's data and any other known SRS. Documentation on how to create at atlas hub can be found at http://code.google.com/p/incf-dai/wiki/HowToCreateAHub. The documentation points to general code libraries and hubs implemented within the project, which can be leveraged by software developers in creating new atlas hubs. The software, including WaxML schema, libraries, and coding examples is available at http://code.google.com/p/incf-dai, and can be used by developers wishing to build on any part of DAI. If resources allow in the future, we would create additional



Location:	x: 521.00, y: 582.00, z: 176.00
Hub	ABA
Code	STRd
Description	Striatum dorsal region
Hub	UCSD
Code	CPu
Description	caudate putamen striatum
Hub	WHS
Code	Str
Description	Striatum

FIGURE 9 | Querying DAI resources using POI-based requests in WIB. (A) Web Image Browser (WIB), illustrates how one can query the different atlases from a user-selected POI. As the user browses to a location of interest in the dataset and selects a POI for query, a menu appears showing registered atlas services and functions offered from each hub. Items in the menu invoke POI-based service functions, which return the requested information to the user. The outlines of structures from a reference atlas aid the user during navigation. **(B)** Example query results showing structure names from several atlases, gene correlation map served by Allen Brain Atlas, and spatially registered images near the POI served by CCDB.

tools to more easily implement an atlas hub, at least for certain data types.

CONCLUSIONS AND FUTURE WORK

Today's neuroscientist is quite familiar with using interactive online maps to access diverse information from different sources. Tools like Google Maps are appealing because they serve as gateways to enormous amounts of spatially-registered information. This type of functionality, if available in the realm of neuroscience, would appeal to researchers, as everything is tied to "where in the brain" and relating different data by brain location would greatly facilitate our ability to do rigorous, and unique quantitative analyses (Carson et al., 2005; Kovacević et al., 2005; Christiansen et al., 2006; Leergaard and Bjaalie, 2007; Lein et al., 2007; Ma et al., 2008; Aggarwal et al., 2009; Ng et al., 2009; Chuang et al., 2011). Atlas projects of the Allen Brain Institute are a great example of what is possible when this kind of information is put within the context of spatial maps. Ideally, all neuroscience data would be presented within an accessible spatial framework such as this in order to facilitate our ability to find, analyze, and integrate diverse information. However, given multiple reference atlases developed with different functionality, data types, and spatial and semantic conventions, opportunities for researchers to easily access and integrate data from many of them, remain limited. Even more difficult, is the ability for most researchers to place their own data into a compatible spatial framework for comparison and analysis. This is becoming an acute problem with new techniques for 3D brain imaging such as microCT and

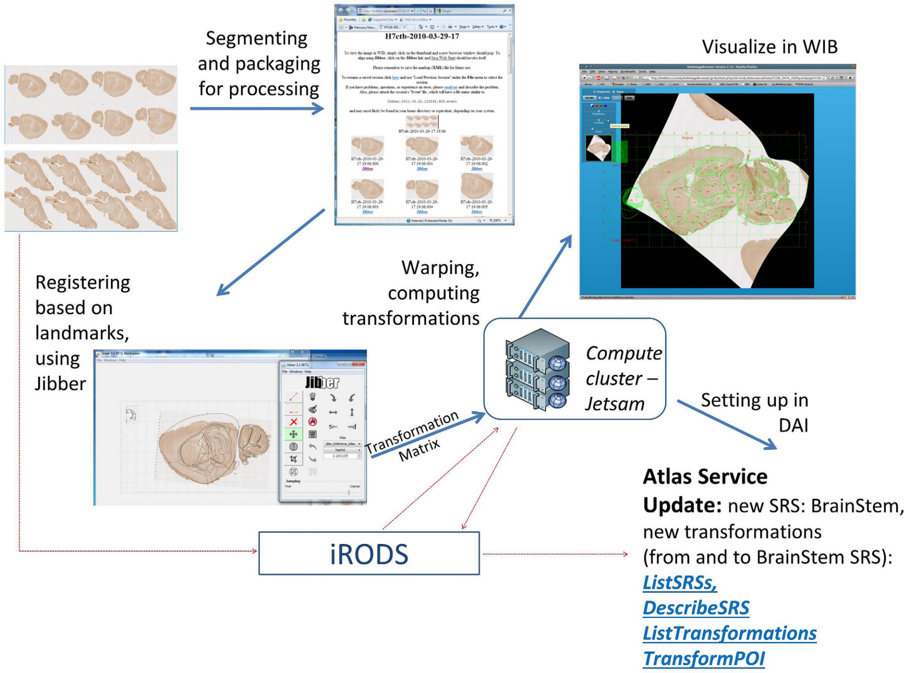

Segmenting and packaging for processing
H7ctb-2010-03-29-17
Visualize in WIB
Registering based on landmarks, using Jibber
Warping, computing transformations
Compute cluster – Jetsam
Transformation Matrix
Setting up in DAI
iRODS
Atlas Service
Update: new SRS: BrainStem, new transformations (from and to BrainStem SRS):
ListSRSs,
DescribeSRS
ListTransformations
TransformPOI

FIGURE 10 | Main steps of the atlas registration workflow for collections of 2D images. The example images are from a study of innervation and genetic similarity in brainstem (Matthews, 2012). The images are segmented, packaged together and uploaded to INCF DataSpace. Subsequent steps include generation of an image gallery page, aligning individual images in the gallery with target reference plates (using Jibber), generating thin plain spline transformations, generating warped images (using Jetsam), generating and updating a new SRS description (called BrainStem) and forward and inverse transformations between the new SRS and the target reference atlas (in this case, the ABA reference atlas). Once the user has registered their data, they can identify areas of interest in their datasets and apply information from other Atlas Hubs to their data (e.g., what structure is found at this location in space in the Allen Brain Atlas). More analytic capabilities are also possible, but these are not currently offered by the INCF Digital Atlasing Program.

methodologies for whole-brain fluorescent imaging (Susaki et al., 2014).

The purpose of this project is to fill the digital atlasing needs of neuroscientists who lack the resources to explore the rapidly growing collections of multidimensional atlas data based on brain location, compare their data with canonical atlases, or publish their data and make it accessible to others via spatial queries. Creating a data-rich and uniform spatial integration framework for atlas sources is challenging because of diversity across reference atlases, data types, and technologies, in addition to the lack of native spatial query functionality of atlas publishers. Thus, our solution has been to create a flexible and extensible framework that accepts existing resources, offers them formal descriptions, in addition to translations and spatial data exchange mechanisms between them.

The INCF-DAI framework addressed these atlas data integration challenges by developing information models for spatial references systems (SRS) in mouse brain; creating web-accessible registries of SRS and coordinate transformations between them, proposing a standard markup language for encoding SRS, and transformations. It offers the ability to query based on spatial location anatomic features and other common atlas constructs (returned via WaxML) through a system of atlas web services that communicate location information between atlas sources and clients. These components became the backbone of the prototype SOA for brain atlas data, which has been implemented via a collection of atlas hubs hosting web services, service metadata catalogs, central discovery portal, and a collection of atlas clients that use the services to perform coordinate transformations or retrieve information for a given POI. Since a broader consensus about community spatial integration frameworks for the brain is yet to emerge, a key requirement for the infrastructure prototype has been flexibility and extensibility of the specifications and their ability to incorporate different implementations of related functions.

This work demonstrated the power of leveraging spatial information integration resources that have been developed and standardized in other disciplines with longer history of managing and exchanging spatial location information. Reusing international standards for the description of spatial features such as GML, and spatial processing functions such as WPS, allowed us to streamline architecture development and create a more robust and maintainable system leveraging open source standards-compliant software. In addition, this helped us better understand the specifics of spatial representation and spatial information processing for brain data as compared to spatial descriptions used at the earth scales.

There are a number of challenges and limitations of the infrastructure prototype that should be addressed in future work. Ideally, we would be able to extend WHS and DAI approaches to other developmental phases and species, and fully explore the potential of spatial data integration. Relating information across phases and species would help address key research issues that underlay the use of all animal models of human neurological disorders. In addition, we would also like to create additional tools, resources, and documentation that reduces the effort needed for researchers to add their data to this framework, or to take advantage of it for their own analysis purposes.

More technical desired additions to the DAI include:

- Formal modeling of coordinate transformations that can accommodate different types of atlas references spaces.
- Consistent assessment of performance of coordinate transformations between atlas spaces, in particular evaluating quality of transformations and chains of transformations;
- Incorporating multiple ways of representing location in the brain (by coordinates, by anatomic feature name, by a collection of location rules, i.e., statements that include anatomic features and spatial relationships), and making such representations interoperable. This would be extremely useful for extending DAI to different developmental phases and species, where relating information by coordinates would be unreliable.
- Extending POI-based data exchanges to exchanging information for regions-of-interest, trajectories (along certain paths), transects, etc.
- Building community consensus about common data representation and functionality associated with atlases and further standardizing atlas services.

The latter typically requires significant time, effort and a formal and transparent process involving both neuroscientists and IT experts, which includes several phases: from identifying areas for standardization, to community review of proposed standards, pilot implementations and interoperability experiments, and to adoption and standards management. We believe that addressing atlas data integration challenges in a consistent manner, moving toward best practices and, eventually, community standards for atlas data representation and exchange, allows neuroscientists to more easily share data in a common spatial framework. This in turn, greatly increases accessible data and has the potential to facilitate data analysis, comparison, cross-validation, and integration across disciplines, developmental stages, and species. The work described in this paper offers first steps toward tackling many of the hurdles to sharing spatially-tied data as well as a framework that can be shaped and expanded by the research community.

ACKNOWLEDGMENTS

This work was conducted within the DAI Task Force of the INCF Digital Atlasing Program. Support from INCF over 2010–2012 for the development of DAI components is gratefully acknowledged. Key programmers are: Asif Memon (atlas services), Stephan Lamont (spatial registration workflow), David Valentine (WaxML schemas), David Little (service development and testing framework, documentation), and Raphael Ritz (Python API). We are especially grateful to members of the INCF DAI task force (http://wiki.incf.org/mediawiki/index.php/DAI_TF_People) for useful discussions of the atlasing infrastructure design, review and testing of atlas services, and development of client applications.

REFERENCES

Aggarwal, M., Zhang, J., Miller, M. I., Sidman, R. L., and Mori, S. (2009). Magnetic resonance imaging and micro-computed tomography combined atlas of developing and adult mouse brains for stereotaxic surgery. *Neuroscience* 162, 1339–1350. doi: 10.1016/j.neuroscience.2009.05.070

Avants, B. B., Tustison, N. J., Song, G., Cook, P. A., Klein, A., and Gee, J. C. (2011). A reproducible evaluation of ANTs similarity metric performance in brain image registration. *Neuroimage* 54, 2033–2044. doi: 10.1016/j.neuroimage.2010.09.025

Bakker, R., Larson, S. D., Strobelt, S., Hess, A., Wojcik, D., Majka, P., et al. (2010). Scalable brain atlas: from stereotaxic coordinate to delineated brain region. *Front. Neurosci. Conference Abstract: Neuroinformatics 2010*. doi: 10.3389/conf.fnins.2010.13.00028

Baldock, R. A., Bard, J. B. L., Burger, A., Burton, N., Christiansen, J., Feng, G., et al. (2003). EMAP and EMAGE A framework for understanding spatially organized data. *Neuroinformatics* 1, 309–325. doi: 10.1385/NI:1:4:309

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., et al. (2007). NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucleic Acids Res.* 35, D760–D765. doi: 10.1093/nar/gkl887

Bjaalie, J. G. (2002). Localization in the brain: new solutions emerging. *Neuroscience* 3, 322–325. doi: 10.1038/nrn790

Boline, J., Lee, E. F., and Toga, A. W. (2008). Digital atlases as a framework for data sharing. *Front. Neurosci.* 2, 100–106. doi: 10.3389/neuro.01.012.2008

Carson, J. P., Ju, T., Lu, H. C., Thaller, C., Xu, M., Pallas, S. L., et al. (2005). A Digital atlas to characterize the mouse brain transcriptome. *PLoS Comput. Biol.* 1:e41. doi: 10.1371/journal.pcbi.0010041

Christiansen, J. H., Yang, Y., Venkataraman, S., Richardson, L., Stevenson, P., Burton, N., et al. (2006). EMAGE: a spatial database of gene expression patterns during mouse embryo development. *Nucleic Acids Res.* 34, D637–D641. doi: 10.1093/nar/gkj006

Chuang, N., Mori, S., Yamamoto, A., Jiang, H., Ye, X., Xu, X., et al. (2011). An MRI-based atlas and database of the developing mouse brain. *Neuroimage* 54, 80–89. doi: 10.1016/j.neuroimage.2010.07.043

David, P. A., and Greenstein, S. M. (1990). The economics of compatibility standards: an introduction to recent research. *Econ. Innov. New Techn.* 1, 3–42. doi: 10.1080/10438599000000002

Erl, T. (2005). *Service-Oriented Architecture*. Vol. 8. New York, NY: Prentice Hall.

Hawrylycz, M. J., Baldock, R. A., Burger, A., Hashikawa, T., Johnson, G. A., Martone, M., et al. (2011). Digital atlasing and standardization in the mouse brain. *PLoS Comput. Biol.* 7, 2–7. doi: 10.1371/annotation/22c5808a-56cf-46e5-ba1b-456e838a5428

Hawrylycz, M. J., Boline, J., Burger, A., Hashikawa, T., Johnson, G. A., Martone, M., et al. (2009). The INCF digital atlasing program: report on digital atlasing standards in the rodent brain. *Nat. Preced.* doi: 10.1038/npre.2009.4000.1

Hof, P. R., Young, W. G., Bloom, F. E., Belichenko, P. V., and Cello, M. R. (2000). *Comparative Cytoarchitectonic Atlas of the C57BL/6 and 129/Sv Mouse Brains*. Amsterdam: Elsevier.

Johnson, G. A., Badea, A., Brandenburg, J., Cofer, G., Fubara, B., Liu, S., et al. (2010). Waxholm space: an image-based reference for coordinating mouse brain research. *Neuroimage* 53, 365–372. doi: 10.1016/j.neuroimage.2010.06.067

Johnson, G. A., Calabrese, E., Badea, A., Paxinos, G., and Watson, C. (2012). A multidimensional magnetic resonance histology atlas of the wistar rat brain. *Neuroimage* 62, 1848–1856. doi: 10.1016/j.neuroimage.2012.05.041

Josuttis, N. (2007). *SOA in Practice: The Art of Distributed System Design*. O'Reilly Media, Inc.

Kovacević, N., Henderson, J. T., Chan, E., Lifshitz, N., Bishop, J., Evans, A. C., et al. (2005). A three-dimensional MRI atlas of the mouse brain with estimates of the average and variability. *Cereb. Cortex* 15, 639–645. doi: 10.1093/cercor/bhh165

Larson, S. D., Aprea, C., Martinez, J., Little, D., Astakhov, V., Kim, H. S., et al. (2010). An open Google Earth for neuroinformatics: the whole brain catalog. *Front. Neurosci. Conference Abstract: Neuroinformatics 2010*. doi: 10.3389/conf.fnins.2010.13.00137

Lee, D., Ruffins, S., Ng, Q., Sane, N., Anderson, S., and Toga, A. W. (2010). MBAT: a scalable informatics system for unifying digital atlasing workflows. *BMC Bioinformatics* 11:608. doi: 10.1186/1471-2105-11-608

Leergaard, T. B., and Bjaalie, J. G. (2007). Topography of the complete corticopontine projection: from experiments to principal Maps. *Front. Neurosci.* 1, 211–223. doi: 10.3389/neuro.01.1.1.016.2007

Lein, E. S., Hawrylycz, M. J., Ao, N., Ayres, M., Bensinger, A., Bernard, A., et al. (2007). Genome-wide atlas of gene expression in the adult mouse brain. *Nature* 445, 168–176. doi: 10.1038/nature05453

Ma, Y., Smith, D., Hof, P. R., Foerster, B., Hamilton, S., Blackband, S. J., et al. (2008). *In Vivo* 3D digital atlas database of the adult C57BL/6J mouse brain by magnetic resonance microscopy. *Front. Neuroanat*. 2:1. doi: 10.3389/neuro.05.001.2008

MacKenzie-Graham, A., Jones, E. S., Shattuck, D. W., Dinov, I. D., Bota, M., and Toga, A. W. (2003). The Informatics of a C57BL/6J mouse brain atlas. *Neuroinformatics* 1, 397–410. doi: 10.1385/NI:1:4:397

MacKenzie-Graham, A., Lee, E. F., Dinov, I. D., Bota, M., Shattuck, D. W., Ruffins, S., et al. (2004). A multimodal, multidimensional atlas of the C57BL/6J mouse brain. *J. Anat.* 204, 93–102. doi: 10.1111/j.1469-7580.2004.00264.x

Martone, M. E., Gupta, A., and Ellisman, M. H. (2004). E-neuroscience: challenges and triumphs in integrating distributed data from molecules to brains. *Nat. Neurosci*. 7, 467–472. doi: 10.1038/nn1229

Matthews, D. W. (2012). *The Architecture of the Mouse Trigeminal-Facial Brainstem?: Disynaptic Circuitry, Genomic Organization, and Follicle Mechanics*. Ph.D. Dissertation, UC San Diego: b7625979.

Ng, L., Bernard, A., Lau, C., Overly, C. C., Dong, H. W., Kuan, C., et al. (2009). An anatomic gene expression atlas of the adult mouse brain. *Nat. Neurosci.* 12, 356–362. doi: 10.1038/nn.2281

Orloff, D. N., Iwasa, J. H., Martone, M. E., Ellisman, M. H., and Kane, C. M. (2013). The cell: an image library-CCDB: a curated repository of microscopy data. *Nucleic Acids Res*. 41, D1241–D1250. doi: 10.1093/nar/gks1257

Papp, E. A., Kjonigsen, L. J., Lillehaug, S., Johnson, G. A., Witter, M. P., Leergaard, T. B., et al. (2013). Volumetric Waxholm Space atlas of the rat brain for spatial integration of experimental image data. *Front. Neuroinform. Conference Abstract: Neuroinformatics 2013*. doi: 10.3389/conf.fninf.2013.09.00003

Paxinos, G. (2004). *The Mouse Brain in Stereotaxic Coordinates*. Amsterdam: Gulf Professional Publishing.

Paxinos, G., and Franklin, K. B. J. (2001). *The Mouse Brain in Stereotaxic Coordinates (Deluxe Edition), 2nd Edn*. San Diego, CA: Academic Press.

Paxinos, G., Halliday, G., Watson, C., Koutcherov, Y., and Wang, H. (2007). *Atlas of the Developing Mouse Brain*. San Diego, CA: Academic Press.

Paxinos, G., and Watson, C. (1998). *The Rat Brain in Stereotaxic Coordinates*. 4th Edn. San Diego, CA: Academic Press.

Paxinos, G., and Watson, C. (2009). *Chemoarchitectonic Atlas of the Mouse Brain*. San Diego, CA: Academic Press.

Portele, C. (2007). *OpenGIS Geography Markup Language (GML) Encoding Standard*. Wayland, MA: Open Geospatial Consortium. Rep. No. 3.2 1.

Ruffins, S. W., Lee, D., Larson, S. D., Zaslavsky, I., Ng, L., and Toga, A. W. (2010). MBAT at the Confluence of Waxholm Space. *Front. Neurosci. Conference Abstract: Neuroinformatics 2010*. doi: 10.3389/conf.fnins.2010.13.00132

Spellman, P., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., et al. (2002). Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.* 3, research0046.1–research0046.9. doi: 10.1186/gb-2002-3-9-research0046

Susaki, E., Tainaka, K., Perrin, D., Kishino, F., Tawara, T., Watanabe, T., et al. (2014). Whole-brain imaging with single-cell resolution using chemical cocktails and computational analysis. *Cell* 157, 726–739. doi: 10.1016/j.cell.2014.03.042

Swanson, L. (1998). *Brain Maps: Structure of the Rat Brain, 2nd Edn*. Amsterdam: Elsevier.

Toga, A. W. (2002). Neuroimage databases: the good, the bad and the ugly. *Nat. Rev. Neurosci*. 3, 302–309. doi: 10.1038/nrn782

Van den Brink, L., Portele, C., and Vretanos, P.A. (2012). "OpenGIS Implementation Standard Profile 10-100r3: Geography Markup Language (GML) simple features profile (with Corrigendum)," in *Technical Report* (Open Geospatial Consortium Inc.). Available online at: http://www.opengeospatial.org/standards/gml (Accessed January 20, 2014).

West, J. (2007). "The economic realities of open standards: black, white and many shades of gray," in *Standards and Public Policy*, eds S. Greenstein and V. Stango (Cambridge: Cambridge University Press), 87–122.

Zakiewicz, I. M., van Dongen, Y. C., Leergaard, T. B., and Bjaalie, J. G. (2011). Workflow and atlas system for brain-wide mapping of axonal connectivity in rat. *PLoS One* 6:e22669. doi: 10.1371/journal.pone.0022669

Zaslavsky, I., He, H., Tran, J., Martone, M. E., and Gupta, A. (2004). "Integrating brain data spatially: spatial data infrastructure and atlas environment for online federation and analysis of brain images," in *Proceedings. 15th International Workshop on Database and Expert Systems Applications, 2004* (Zaragoza), 389–393. doi: 10.1109/DEXA.2004.1333505

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 04 February 2014; accepted: 08 August 2014; published online: 12 September 2014.

*Citation: Zaslavsky I, Baldock RA and Boline J (2014) Cyberinfrastructure for the digital brain: spatial standards for integrating rodent brain atlases. Front. Neuroinform. **8**:74. doi: 10.3389/fninf.2014.00074*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

TECHNOLOGY REPORT ARTICLE
published: 03 July 2014
doi: 10.3389/fninf.2014.00065

Data dictionary services in XNAT and the Human Connectome Project

Rick Herrick**, Michael McKay, Timothy Olsen, William Horton, Mark Florida, Charles J. Moore and Daniel S. Marcus

Neuroinformatics Research Group, Department of Radiology, Washington University School of Medicine, St. Louis, MO, USA

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
B. Nolan Nichols, University of Washington, USA
Gregory A. Book, Hartford Hospital, USA

****Correspondence:***
Rick Herrick, Neuroinformatics Research Group, Mallinckrodt Institute of Radiology, Washington University School of Medicine, Campus Box 8225, 4525 Scott Avenue, St. Louis, MO 63119-2451, USA
e-mail: rick.herrick@wustl.edu

The XNAT informatics platform is an open source data management tool used by biomedical imaging researchers around the world. An important feature of XNAT is its highly extensible architecture: users of XNAT can add new data types to the system to capture the imaging and phenotypic data generated in their studies. Until recently, XNAT has had limited capacity to broadcast the meaning of these data extensions to users, other XNAT installations, and other software. We have implemented a data dictionary service for XNAT, which is currently being used on ConnectomeDB, the Human Connectome Project (HCP) public data sharing website. The data dictionary service provides a framework to define key relationships between data elements and structures across the XNAT installation. This includes not just core data representing medical imaging data or subject or patient evaluations, but also taxonomical structures, security relationships, subject groups, and research protocols. The data dictionary allows users to define metadata for data structures and their properties, such as value types (e.g., textual, integers, floats) and valid value templates, ranges, or field lists. The service provides compatibility and integration with other research data management services by enabling easy migration of XNAT data to standards-based formats such as the Resource Description Framework (RDF), JavaScript Object Notation (JSON), and Extensible Markup Language (XML). It also facilitates the conversion of XNAT's native data schema into standard neuroimaging vocabularies and structures.

Keywords: XNAT, ontologies, translations, publishing, human connectome, human computer interaction

INTRODUCTION

XNAT provides a robust and advanced set of tools for searching and filtering the data that it manages (Marcus et al., 2007)[1]. This includes searching for a project or a set of projects, for subjects that meet particular criteria, and for imaging sessions or subject assessments that match a complex set of attributes. Users can join searches across system object types to search on combinations of properties including subject attributes, imaging modality, and assessed demographic or clinical conditions.

This search function was originally created to work with the attributes and properties of the core system data types. This works well enough for a standard XNAT installation, but users often need to add custom field definitions to existing data types and custom data types to represent domain- or project-specific imaging modalities and patient or subject assessments. Further, XNAT's data types and objects are maintained internally in a verbose and fairly complex Extensible Markup Language (XML)-based[2] structure.

In addition, the advanced search functions in XNAT are quite technical in their presentation and workflow design, so users need a detailed understanding of the underlying data. This search interface works well for experienced XNAT users, but, it is a significant barrier to users who are knowledgeable about the research domain but unfamiliar with XNAT in general or with the specific data models for a particular project.

This became a critical issue for the Human Connectome Project (HCP) (Van Essen et al., 2012)[3]. The HCP public data distribution site is ConnectomeDB[4], an XNAT instance that delivers curated content to users and leverages the underlying XNAT data structures to define groups of subjects within the overall research population, such as a group of 40 unrelated subjects and a group of 120 (some related) individuals, as well as data sets of particular interest, such as group average resting state connectivity, task fMRI data, and behavioral data scores (Marcus et al., 2013).

The target audience for ConnectomeDB is assumed to be sophisticated in its understanding of the data produced by HCP data acquisition and processing, but, unlike experienced XNAT users, cannot be assumed to have anything other than a basic

[1] XNAT, NRG Lab at the Washington University School of Medicine. http://www.xnat.org.

[2] XML Core Working Group (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition) http://www.w3.org/TR/xml/.

[3] The Human Connectome Project. https://www.humanconnectome.org.

[4] ConnectomeDB. https://db.humanconnectome.org.

level of skill with XNAT's search and retrieval features and little or no familiarity with the general XNAT and HCP-specific data models.

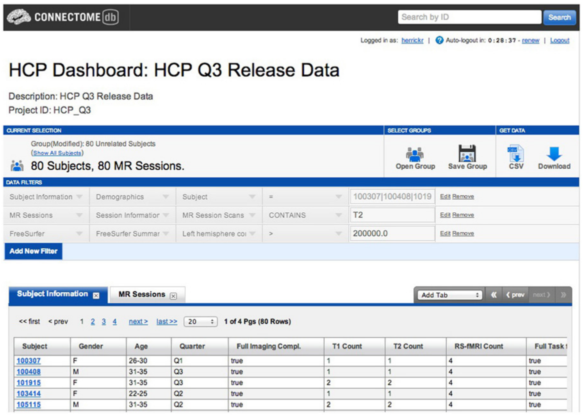
This issue—translating concepts, terminology, and operational paradigms specific to XNAT into vocabularies appropriate to other contexts—is one that the XNAT team has encountered frequently as the field of electronic medical imaging has grown and research organizations have become more connected. This has led to greater interest in being able to access and reference data from researchers around the world, without the archive and data management software imposing lexical or syntactical barriers to translation. This is the motivation behind INCF efforts to promote metadata and standards for data sharing within the medical imaging research community (Poline et al., 2012). We took the opportunity afforded by our immediate problem to solve the larger problem in front of us: to transform XNAT from a relatively isolated data archiving repository into an adaptable, query-able, and standards-based data sharing, aggregation, and distribution service.

DEVELOPMENT GOALS

One of the key features of ConnectomeDB is the subject dashboard. The dashboard provides an intuitive way to search for subjects based on demographics, clinical assessments, and other relevant data, with textual cues and cumulative specification of search criteria. In a standard XNAT installation, users define custom subject groups by defining reusable queries using the advanced search feature. The requirements for publishing to a more generalized audience required a more textual and intuitive means of achieving the same goal.

The subject dashboard lets users create one or more data filters to identify subject groups of interest (**Figure 1**). Each filter works on some instrument associated with the subjects in the system. These instruments can be directly related to the subject, such as demographics or other subject metadata; extrapolated from subject performance or evaluation, such as performance on cognitive awareness tests, personality evaluations, or physical exams and evaluations; or extracted from imaging data associated with the subject, including types of acquired imaging data, attributes of the data, and processed and secondary capture data.

We needed to make this sort of rich search functionality available to a general audience in a readily understandable way. The *IBM Dictionary of Computing* defines a data dictionary as a "repository of information about data such as meaning, relationships to other data, origin, usage, and format[5]." In this reading, a data dictionary provides a layer of meaning and context above simple presentation of the available data types and attributes in the system. This enables the association of natural-language synonyms, descriptions, and relational definitions onto the data attributes and structures in the XNAT data store. The initial development goal thus became creating a framework that could enable the definition of the entities within a specific dictionary instance, but remain general in its design, allowing the framework to be re-purposed for other applications leveraging the advantages of a well-defined contextual and descriptive structure for their particular object structure, attributes, and relationships.



HCP Dashboard: HCP Q3 Release Data

Description: HCP Q3 Release Data
Project ID: HCP_Q3

Group(Modified): 80 Unrelated Subjects
(Show All Subjects)
80 Subjects, 80 MR Sessions.

Subject Information	Demographics	Subject	=	100307|100408|1019	Edit Remove
MR Sessions	Session Informatior	MR Session Scans	CONTAINS	T2	Edit Remove
FreeSurfer	FreeSurfer Summar	Left hemisphere coı	>	200000.0	Edit Remove

1 of 4 Pgs (80 Rows)

Subject	Gender	Age	Quarter	Full Imaging Compl.	T1 Count	T2 Count	RS-fMRI Count	Full Task f
100307	F	26-30	Q1	true	1	1	4	true
100408	M	31-35	Q3	true	1	1	4	true
101915	F	31-35	Q3	true	2	2	4	true
103414	F	22-25	Q2	true	1	1	4	true
105115	M	31-35	Q2	true	2	2	4	true

FIGURE 1 | HCP subject dashboard.

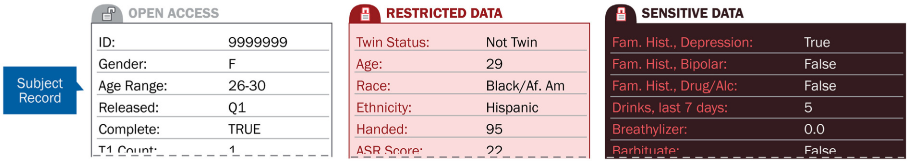
In the course of ConnectomeDB development, a second distinct but related application of data dictionary functionality arose. The subject dashboard allows users to define groups of subjects by various attributes of the subjects. Although measures have been taken to protect the identities of the subjects in the HCP study—anonymized subject identifiers, five-year age bands, and other abstractions from direct personal characteristics—the richness of the available demographic and physical data on each subject raises the possibility that users could piece together enough details about subjects that the data could be considered individually identifiable.

The administrative solution for this was to limit the types of subject data available to standard ConnectomeDB users. Although all users of the site must consent to a data-use agreement to access even the "open access" data, users who require access to data deemed to be especially sensitive must accept an additional data-use agreement with more stringent terms.

Accepting and recording acceptance of restricted data-use agreements is just the first step. To implement tiered access to subject data, the system needs to define what those tiers are, as well as what types of subject attributes, assessment instruments, imaging data, etc., are available to each tier. The data dictionary provides a convenient means of defining the tier restriction of entities within the system. The scope and comprehensiveness of the system object hierarchy allows for a great deal of flexibility in how data access may be restricted by tier. This means that ConnectomeDB can use a broad brush to shield entire categories of objects from access, but in other cases can limit access to only a single attribute on a particular item. The data tiers currently defined for ConnectomeDB are:

- Open access.
- Restricted access, for attributes that contain data that could be potentially identifying.
- Sensitive access, which includes data like drug screening and family history of mental illness.
- Confidential access, which is inappropriate for any release, such as record of criminal behavior.

[5] Dictionary of IBM and Computing Terminology. http://www-03.ibm.com/ibm/history/documents/pdf/glossary.pdf.



OPEN ACCESS	
ID:	9999999
Gender:	F
Age Range:	26-30
Released:	Q1
Complete:	TRUE
T1 Count:	1

RESTRICTED DATA	
Twin Status:	Not Twin
Age:	29
Race:	Black/Af. Am
Ethnicity:	Hispanic
Handed:	95
ASR Score:	22

SENSITIVE DATA	
Fam. Hist., Depression:	True
Fam. Hist., Bipolar:	False
Fam. Hist., Drug/Alc:	False
Drinks, last 7 days:	5
Breathylizer:	0.0
Barbituate:	False

FIGURE 2 | Access tiers secure subject attribute data based on the level of user data-use agreements.

Figure 2 shows how the attributes of a resource, such as data collected on a particular subject, are categorized into access tiers. These tiers can then be used to allow access to particular types of information only to users approved to view the data in that tier.

IMPLEMENTATION

FRAMEWORKS AND PLATFORMS

Creating an XNAT-specific framework allowed us to complete development and testing of the new service within the tight development timeframe, while also achieving the goals of fitting within our existing development technologies and semantic context, and maintaining our ability to continue to release XNAT with minimal external dependencies.

The XNAT data dictionary was written as an abstract service definition, specifying the contract between the service and its clients while still allowing for flexibility in the implementation. Recent development work on the core XNAT platform has relied primarily on a configuration framework that enables switching between different implementations of a service definition based on the requirements and resources available to a particular deployment—a concept known as *dependency injection*. **Figure 3** shows how the data dictionary can be accessed via a conceptual interface that abstracts the functionality, while the actual core back-end functionality can be implemented in a number of different ways.

The contract for the XNAT data dictionary service is defined through a few simple interfaces and entity definitions, while the initial implementation of that service is defined in a concrete implementation class. This leaves the path open to creating future implementations that do utilize the power of stand-alone triple-store servers like Apache Jena, but with the advantage of an API that fits more easily into the context of the XNAT development framework.

DETAILS

The data dictionary service is implemented as a Java library, with an abstract interface that defines the basic operations of the service. These service operations are performed on a small set of entity types, defined as Java beans, simple data objects that amount to a list of object properties along with functions to retrieve the value of those properties and, in many cases, set the value for those properties. These Java beans are then made available in XNAT through direct calls to the library from within XNAT and through an extension of the XNAT web services API for access from the client-side user interface. A specific instance of a data dictionary is defined in a JavaScript Object Notation (JSON)-formatted[6] configuration file that is read by the dictionary service on initialization.

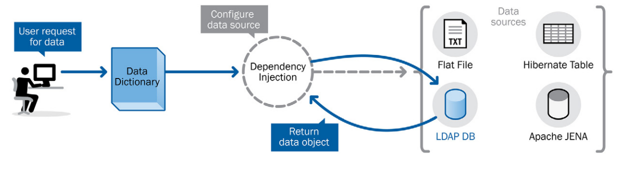


FIGURE 3 | Dependency injection allows service access while providing flexibility in implementation.

NODES

The basic building block of the dictionary elements is the node entity. Nodes include a number of core properties that are used by all objects in the dictionary service, shown in **Table 1**.

The service includes three specific node types: categories, assessments, and attributes. These nodes are organized in a simple hierarchy and contain additional hierarchy-level specific properties.

A category maps to a research domain and contains a group of assessments. This is the highest level of organization and so encompasses the largest conceptual groupings in the data dictionary.

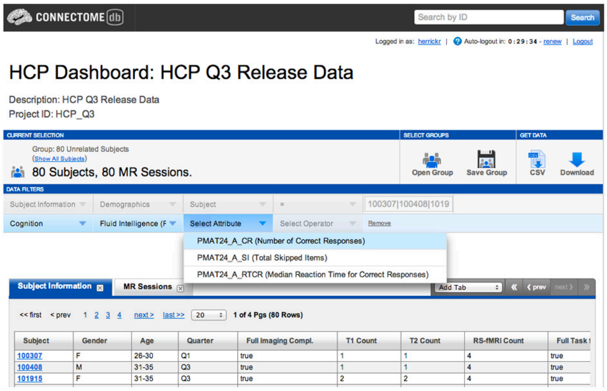
An assessment node defines a set of related observations about a subject. Assessments are any discreet collection of data that evaluates or describes some aspect of the assessed subject. An assessment defined in the data dictionary generally maps to a standard XNAT or HCP-specific data object, such as subject demographics, clinical assessment, imaging data, and so on. However, mapping is not defined at the assessment level but at the attribute level, so it would be possible to create a data dictionary assessment that actually comprises multiple data types. For example, a clinical assessment type might combine an attribute from direct clinical observations such as MMSE or an IQ assessment with attributes from a genetic or demographic assessment.

An assessment is normally defined within XNAT by its relationship to its project and subject, but the data dictionary service adds an extra definition for the assessment's category. This allows the data dictionary service to group similar assessments for conceptual and organizational purposes. This relationship to a category is the primary and only metadata contained in the assessment dictionary type outside of the base node attributes.

[6] JSON is an open standard format with no canonical definition: http://en.wikipedia.org/wiki/JSON.

Table 1 | Node attributes.

Property	Description
Name	The name of the node.
Description	A description of the contents of the node.
Position	The suggested position of the node relative to its siblings.
Column header	A shortened version of the node name for display in column headers and restricted display areas.
Projects	A list of XNAT project identifiers with which the node is associated. Not all entities within the data dictionary may be applicable to all projects. For example, as new data models are added to data sets, particular nodes may only be applicable to the later releases that include those models.
Tier	Indicates the security access level required to view or query on the node.


CONNECTOME db
Search by ID
Search
Logged in as: herrickr | Auto-logout in: 0:29:34 - renew | Logout

HCP Dashboard: HCP Q3 Release Data

Description: HCP Q3 Release Data
Project ID: HCP_Q3

Group: 80 Unrelated Subjects
(Show All Subjects)
80 Subjects, 80 MR Sessions.

Subject Information | Demographics | Subject | = | 100307|100408|1019
Cognition | Fluid Intelligence (F | Select Attribute | Select Operator | Remove

PMAT24_A_CR (Number of Correct Responses)
PMAT24_A_SI (Total Skipped Items)
PMAT24_A_RTCR (Median Reaction Time for Correct Responses)

<< first < prev 1 2 3 4 next > last >> 20 1 of 4 Pgs (80 Rows)

Subject	Gender	Age	Quarter	Full Imaging Compl.	T1 Count	T2 Count	RS-fMRI Count	Full Task f
100307	F	26-30	Q1	true	1	1	4	true
100408	M	31-35	Q3	true	1	1	4	true
101915	F	31-35	Q3	true	2	2	4	true

FIGURE 4 | Category, assessment, and attributes are browse-able through the ConnectomeDB dashboard.

Table 2 | Defining category node-type properties in data-dictionary.

Property	Description
Category	Every assessment belongs to a single category. This property is a key to the category name.

Example JSON	{
	"name": "Cognition",
	"columnHeader": "Cognition",
	"description": "Cognition",
	"tier": 0,
	"position": 4,
	"projects": ["HCP_Q1","HCP_Q2", "HCP_Q3","HCP_Q3_RST"]
	}

Table 3 | Defining assessment node-type properties in data-dictionary.

Property	Description
Category	Every assessment belongs to a single category. This property is a key to the category name.

Example JSON	{
	"position": 1,
	"category": "Cognition",
	"name": "Episodic Memory (Picture Sequence Memory)",
	"columnHeader": "Episodic Memory",
	"description": "",
	"tier": 0,
	"projects": ["HCP_Q2","HCP_Q3"]
	}

An attribute node defines a particular evaluation or observation. Attributes are the various data points and measurements that make up the content of an assessment. Each attribute in the data dictionary maps directly to a field contained in a core XNAT or HCP-specific data type. Attribute definitions also provide a number of ways to help users enter valid values for the attribute. For example, a list of valid comparison operators restricts the types of operations users can perform against the value of the attribute, a list of valid attribute values limits the values that can be set for the attribute, and so on. The combination of these various types of user assistance provides unobtrusive guidance and assistance to users when navigating the search function on the subject dashboard. **Table 4** shows the full list of properties on the data dictionary attribute definition.

An example of a full relationship would be the category of Cognition, which includes a number of assessments, such as Fluid Intelligence, which in turn includes a number of attributes, including number of correct responses, total skipped items, and median reaction time for correct responses. The representation of this relationship is illustrated in **Figure 4**.

Each level of data dictionary entity can be defined with a set of attributes recognized by the data dictionary service. **Tables 2–4** describe the unique properties available on each specific node implementation.

This relatively simple hierarchical structure provides a framework for defining, navigating, and, most importantly, searching and querying all of the data in ConnectomeDB in a very user-friendly manner. The specific instances of categories, assessments, and attributes map very closely to the complex data type definitions in XNAT that the advanced search functions are designed for, but the descriptive and contextual metadata defined in the data dictionary makes the search functionality closer to natural language.

Once we have defined the data dictionary entries, the configuration is deployed to the server. At that point, the categories, assessments, and attributes are available through the XNAT data dictionary service and can be accessed by any other service that wants to use them. The means by which client services access the data dictionary depends on the relationship to the XNAT server.

For internal XNAT services, that is, services that execute on the same application server and within the same process space as the data dictionary service, there is a programmatic service that can be accessed through XNAT's standard application context. This allows querying of the various entities within the data dictionary, translation of data dictionary entities into addressable XNAT data

Table 4 | Defining attribute node-type properties in data-dictionary.

Property	Description
Category	Corresponds to the category that owns the attribute.
Assessment	Corresponds to the assessment that owns the attribute.
XSI Type	Maps to an XNAT data type that can contain the actual instances of subject data for the attribute.
Field ID	Indicates the field on the XNAT data type for this particular attribute.
Display name	Provides a readable name for the attribute.
Operators	Indicates the types of comparison operations that can be performed on values for the attribute.
Values	Indicates values for the attribute when the value is restricted to a list, e.g. M or F for gender, true or false to indicate whether a full study protocol has been completed, and so on.
Validation	A regular expression that can be used to validate data entered by the user.
Validation message	A validation message to be displayed when data entered by the user has failed to match the validation regular expression.
Watermark	Suggestive text displayed in free-form text entry boxes to assist users in understanding the proper format for entering data.

Example JSON	{
	   "name": "COG_PIC_SEQ_USCR",
	   "category": "Cognition",
	   "assessment": "Episodic Memory (Picture Sequence Memory)",
	   "fullDisplayName": "NIH Toolbox Picture Sequence \\
	                       Memory Test... ",
	   "dictType": "Float",
	   "validationMessage": "Picture Sequence Unadjusted \\
	                         must be a... ",
	   "validation": "^[-+]?[0-9]*[.]?[0-9]+$",
	   "columnHeader": "Picture Sequence Unadjusted",
	   "operators": {
	      "=": "=",
	      "!=": "NOT =",
	      "<": "<",
	      ">": ">"
	   },
	   "watermark": "usually 70-140",
	   "xsiType": "hcp:ToolboxData",
	   "fieldId": "COG_PIC_SEQ_USCR",
	   "position": "1",
	   "description": "The Picture Sequence Memory Test is \\
	                   a measure... ",
	   "tier": 0,
	   "projects": ["HCP_Q2","HCP_Q3"]
	}

types and attributes, and rendering of the data dictionary into various data interchange formats (currently the data dictionary service supports only JSON as an interchange format, but future development efforts will extend the available formats to allow for integration with non-XNAT systems and querying tools).

The data dictionary service also provides a Web service that allows services and tools outside of XNAT full access to the metadata and structures in the data dictionary. This provides a means to explore the structures that are defined in the data dictionary. For example, the search filter function shown in **Figure 3** is essentially a means of browsing through the data dictionary entities representing the categories of assessments and attributes. It also functions as a translational layer from a conceptual entity—such as a particular attribute and potential values for that attribute—to an actionable data object within the XNAT system. In this view, the XNAT data dictionary service works as a translational tool: rendering technical or domain-specific terminology and nomenclature into formats or language more suited to a particular audience or type of user.

APPLICATION

The first and most obvious usage of the data dictionary's web services API is in the various user interface elements on ConnectomeDB. The search filters in **Figure 3** act as a browser for the various data dictionary entities. Once a user has composed a search operation of one or more filters, the query parameters may be checked against validation expressions associated with the attributes. And once the query has been successfully validated, the data dictionary translates the specified data dictionary attributes into XNAT-specific search queries that be run against the server's data store. This makes it much easier for researchers to work with language and concepts with which they are well acquainted to leverage the functionality of XNAT's search and data retrieval services.

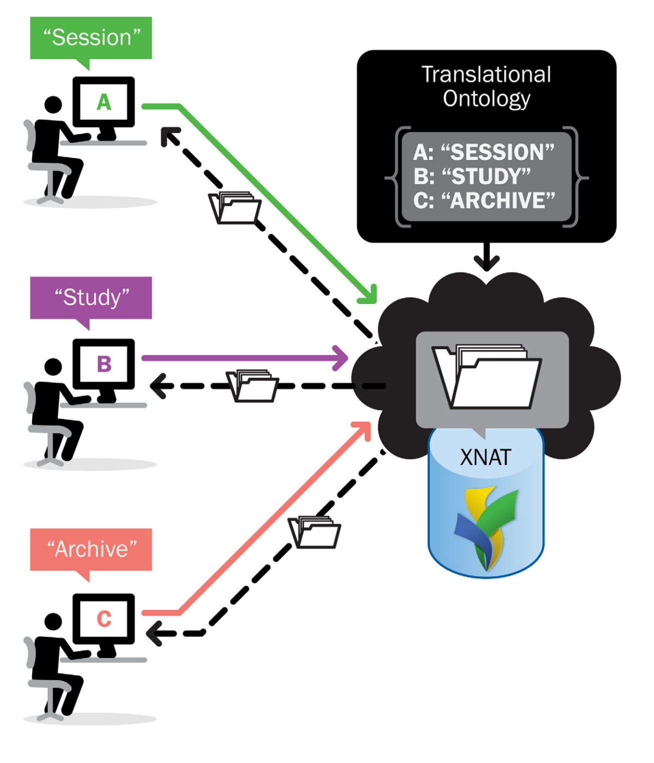


FIGURE 5 | Translation of different vocabularies to XNAT entities.

But one of the XNAT development team's far-reaching goals is seamless integration with medical imaging and electronic data capture systems across research organizations. The XNAT data dictionary REST service can help achieve this goal precisely through the same translational function that allows for greater ease of use for human users. Many applications have differing terminology and particular means of structuring, storing, and retrieving data and metadata. There are groups working to standardize the terminology and ensure interoperability amongst those applications, since the end goal for research data services is almost always to make the data as available as possible to the greater research community. It is at that point that these differences in structure and verbiage need to be negotiated and bridged. The XNAT data dictionary services provide a flexible means of mapping these other vocabularies and structures onto XNAT's internal structure, as shown in **Figure 5**.

DEPLOYMENT

The XNAT data dictionary service was initially deployed with the HCP Q3 data release[7]. The service as deployed on ConnectomeDB comprises the following components:

- The core data dictionary service definition and implementation, which provides the underlying metadata persistence and retrieval service.
- The data dictionary REST API, which provides access to the data dictionary service via HTTP.
- A data dictionary search service that returns XML used to build and decorate the tables containing the results of searches based on criteria defined in the data dictionary.

The code and configuration files for these can be found in the Mercurial repository for ConnectomeDB customizations at https://bitbucket.org/hcp/db_builder_customizations. The following sections reference particular code components and configurations from this repository.

CORE DATA DICTIONARY SERVICE

The core data dictionary service is defined by the **DataDictionaryService** interface and implemented for this deployment in the **SimpleDataDictionaryService** class. Upon instantiation, the **SimpleDataDictionaryService** loads a statically defined JSON configuration, contained in the **datadictionary-context.xml** configuration file, to construct and manage the system-wide data dictionary.

This simple service has the advantage of being portable and lightweight, requiring no database connection, persistence layer, or transaction management. Its disadvantage is a lack of flexibility and difficulties in maintaining and extending the data dictionary. Given the specific nature of the HCP deployment and the project's well defined study protocol and set of data types, we opted for the simplest implementation at the cost of extensibility. However, as described earlier, the abstraction of the service interface and configurability of service libraries through dependency injection allows for relatively easy switching between different implementations. This will ease the migration of the service to the XNAT platform, which requires more configurability and extensibility than a statically defined library offers.

The core data dictionary service is accessible through direct calls to its API. The classes as currently constituted aren't available as a stand-alone library.

REST API

The REST API is the primary means by which clients of the data dictionary access the service. In the case of ConnectomeDB, service clients consist almost exclusively of authenticated users accessing the data dictionary through their browser as part of a login session on the Web site, but, unlike actual data from the HCP study, the data dictionary service can be accessed without authentication by calling the appropriate URLs directly.

The REST API can be accessed through a number of different URIs into the system (in the following table, all calls are relative to the root of the Web service, which in the case of ConnectomeDB is https://db.humanconnectome.org). In the list below, italicized terms are replaced by specific argument values that indicate what specific data the REST call should return.

/data/services/ddict/*tier*

This function returns a list of data in the specified tier. The ConnectomeDB data dictionary includes only two separate tiers, **categories**, which returns the top-level categories in the data dictionary, and **attributes**, which returns the whole data dictionary

[7] HCP Q3 Data Release Reference. https://www.humanconnectome.org/documentation/Q3.

in a single JSON structure. The categories tier can be used to drill down into specific categories all the way down to the attribute level in an efficient way, while the attributes tier is used to retrieve all data in single operation. The first approach is very efficient in terms of the amount of data retrieved on each separate call to the service, while the second approach is efficient in terms of limiting the number of HTTP calls between the server and the REST client.

/data/services/ddict/*tier*/*category*

This function gets a list of all metadata subordinate to the indicated category in the tier. In the ConnectomeDB implementation, the only tier for which this call is valid is **categories**, since getting all attributes for all categories is synonymous with retrieving the full **attributes** tier. For example, to find all assessments associated with the FreeSurfer category, the appropriate REST URI would be **/data/services/ddict/categories/FreeSurfer**.

/data/services/ddict/*tier*/*category*/*assessment*

This function gets a list of all metadata subordinate to the indicated assessment. For example, to find all attributes associated with the FreeSurfer Volume assessment, the appropriate REST URI would be **/data/services/ddict/attributes/FreeSurfer/Volume**. Note that calling this URI with the **categories** tier effectively ignores any arguments to the right of **category**.

/data/services/ddict/*tier*/*category*/*assessment*/*attribute*

This function gets the metadata associated with a particular attribute. This is an efficient way to retrieve the metadata about a specific attribute when you know the category and assessment to which the attribute belongs. For example, to get the metadata about the cerebral spinal fluid volume measure on the FreeSurfer assessment, the appropriate URI path would be **/data/services/ddict/attributes/FreeSurfer/Volume/_CSF**.

/data/services/ddict/*tier*/*category*/*assessment*/*attribute*/validate/*value*

This REST call provides validation for particular attribute values. Part of the optional metadata that can be associated with an attribute is a regular expression to test a submitted value. If the specified attribute has a validation expression, this method tests the argument for **value** against that regular expression and returns an error if the value doesn't match properly.

DATA DICTIONARY SEARCH SERVICE

This service provides only a single REST function, **/data/services/search/ddict/*category***. This returns all attributes for the indicated category in XML form. This XML is formatted specifically to be used by the search results table on the subject dashboard and is a good example of how the data dictionary can be used to manage the display of user interface elements.

LESSONS LEARNED AND FUTURE DEVELOPMENT

At the start of development of the data dictionary service for the HCP public site, we took the position that our first development efforts would amount to a test run and learning experience to drive future development efforts to create a full-fledged data dictionary and metadata management framework for the XNAT platform. Because release of this framework with ConnectomeDB would not define and restrict future development efforts in core XNAT platform development, we were free to experiment with different approaches to managing the data dictionary, as well as the metadata's relationship to the primary XNAT domain objects such as imaging sessions, research subjects, subject assessors, etc. We also were not restricted by future development goals related to data dictionary implementation, such as supporting export to Resource Description Framework (RDF)[8], triplestore integration, or import and export operations to and from other medical imaging platforms through mapping and translation of XNAT's internal data structures and taxonomies into common vocabularies and taxonomies.

The primary lesson learned from the data dictionary implementation is the significant limitation in implementing the object structure using Java class definitions. In XNAT, building Java code into a deployable application requires a number of build steps. This overhead made the object structure fairly inflexible, with changes to the structure requiring changes to the underlying code, necessitating a new build and deploy of the server software. The next iteration of the data dictionary service will use flexible definitions for the dictionary node definitions themselves. This configurable definition feature will allow an installed XNAT server quickly define data dictionary entity structures along with specific instances of those structures without requiring redeployment of the server and enable developers and administrators of a system to make their data available to other services.

Another lesson, of a more positive nature, is the value of this sort of rich metadata associated with the data types in the system. This was demonstrated when the need arose for restricting access to particular assessment instruments and attributes on those assessments based on the user security level. Adding the restricted access feature was still a significant effort, but was aided significantly by the application of the data dictionary, which was already carrying metadata about the object hierarchy at precisely the level required to add security scoping to data access.

CONCLUSION

In implementing the data dictionary service for ConnectomeDB, we were successful in bridging the gap from XNAT's domain-specific and technical terminology and data structures to the vocabulary and entities that are of real interest to the user who want to perform research rather than learning yet another data management tool. The value and ease of use delivered to the site's users were worth the development resources we committed to the implementation of this feature. We also began the process of presenting XNAT and its data not just as a singular software service, but as a flexible and multi-use data repository. The lessons learned from the process of developing the first version of the XNAT data dictionary service are currently being applied in the planning and development of the next generation of the XNAT imaging platform. Most importantly, we need to simplify the process of defining a data dictionary's structures and mapping to XNAT internal data structures. We also will expand the target

[8] RDF Working Group (2014). RDF Schema 1.1 http://www.w3.org/TR/rdf-schema/.

dialects for translation, with the initial goal of supporting RDF and export of XNAT metadata to standard triplestore services like Apache Jena. By translating from XNAT-specific data to protocols and formats understood by other applications and services, we aim to make XNAT support industry-standard data analysis and mining tools, reporting and visualization frameworks, and modeling and graphics applications. This will allow greater flexibility for researchers to analyze their research data and generated data resources. It will also let XNAT serve as a back-end service for other publishing platforms and research tools. By providing the functions to work as an end-to-end-lifecycle data management tool, we hope to help the research community achieve its core goal of converting basic science into completed research.

REFERENCES

Marcus, D. S., Harms, M. P., Snyder, A. Z., Jenkinson, M., Wilson, J. A., Glasser, M. F., et al. (2013). Human Connectome Project informatics: quality control, database services, and data visualization. *Neuroimage* 80, 202–219. doi: 10.1016/j.neuroimage.2013.05.077

Marcus, D. S., Olsen, T. R., Ramaratnam, M., and Buckner, R. L. (2007). The extensible neuroimaging archive toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data. *Neuroinformatics* 5, 11–34. doi: 10.1385/NI:5:1:11

Poline, J. B., Breeze, J. L., Ghosh, S., Gorgolewski, K., Halchenko, Y. O., Hanke, M., et al. (2012). Data sharing in neuroimaging research. *Front. Neuroinform.* 6:9. doi: 10.3389/fninf.2012.00009

Van Essen, D. C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T. E. J., Bucholz, R., et al. (2012). The Human Connectome Project: a data acquisition perspective. *Neuroimage* 62, 2222–2231. doi: 10.1016/j.neuroimage.2012.02.018

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 31 January 2014; accepted: 15 June 2014; published online: 03 July 2014.
Citation: Herrick R, McKay M, Olsen T, Horton W, Florida M, Moore CJ and Marcus DS (2014) Data dictionary services in XNAT and the Human Connectome Project. Front. Neuroinform. ***8****:65. doi: 10.3389/fninf.2014.00065*
This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 07 April 2014
doi: 10.3389/fninf.2014.00035

Sharing privacy-sensitive access to neuroimaging and genetics data: a review and preliminary validation

Anand D. Sarwate[1], Sergey M. Plis[2], Jessica A. Turner[2,3], Mohammad R. Arbabshirani[2,4] and Vince D. Calhoun[2,4]*

[1] Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ, USA
[2] Mind Research Network, Albuquerque, NM, USA
[3] Department of Psychology and Neuroscience Institute, Georgia State University, Atlanta, GA, USA
[4] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Adam Smith, Pennsylvania State University, USA
Sean Randall, Curtin University, Australia

****Correspondence:***
Vince D. Calhoun, The Mind Research Network, 1101 Yale Blvd. NE, Albuquerque, NM 87106, USA
e-mail: vcalhoun@mrn.org

The growth of data sharing initiatives for neuroimaging and genomics represents an exciting opportunity to confront the "small *N*" problem that plagues contemporary neuroimaging studies while further understanding the role genetic markers play in the function of the brain. When it is possible, open data sharing provides the most benefits. However, some data cannot be shared at all due to privacy concerns and/or risk of re-identification. Sharing other data sets is hampered by the proliferation of complex data use agreements (DUAs) which preclude truly automated data mining. These DUAs arise because of concerns about the privacy and confidentiality for subjects; though many do permit direct access to data, they often require a cumbersome approval process that can take months. An alternative approach is to only share data derivatives such as statistical summaries—the challenges here are to reformulate computational methods to quantify the privacy risks associated with sharing the results of those computations. For example, a derived map of gray matter is often as identifiable as a fingerprint. Thus alternative approaches to accessing data are needed. This paper reviews the relevant literature on differential privacy, a framework for measuring and tracking privacy loss in these settings, and demonstrates the feasibility of using this framework to calculate statistics on data distributed at many sites while still providing privacy.

Keywords: collaborative research, data sharing, privacy, data integration, neuroimaging

1. INTRODUCTION

Neuroimaging data has been the subject of many data sharing efforts, from planned large-scale collaborations such as the Alzheimers Disease Neuroimaging Initiative (ADNI) (Jack et al., 2008) and functional biomedical informatics research network (FBIRN) (Potkin and Ford, 2009) (among others) to less-formalized operations such as openfmri.org (Poldrack et al., 2013) and the grass roots functional connectomes project (FCP) with its international extension (INDI) (Mennes et al., 2013). The Frontiers in Neuroinformatics special issue on "Electronic Data Capture, Representation, and Applications in Neuroimaging" in 2012 Turner and Van Horn (2012) included a number of papers on neuroimaging data management systems, several of which provide the research community some access to their data. In many cases, an investigator must agree to a data usage agreements (DUA): they specify who they are, what elements of the data they want, and often what they are planning to do with it. The researcher must agree to abide by arrangements such as not attempting to re-identify the subjects, not re-sharing the data, not developing a commercial application off the data, and so on. These DUAs may be as simple as a one page electronic questionnaire for contact purposes, or a full multi-page form that requires committee review, institutional official review and signatures being faxed back and forth.

The 2012 publication by members of the INCF Task Force on Neuroimaging Datasharing (Poline et al., 2012), specifically on neuroimaging data sharing, reiterated that data should be shared to improve scientific reproducibility and accelerate progress through data re-use. However, the barriers to data sharing that they identified included the well-known problems of motivation (both the ability to get credit for the data collected, as well as the fear of getting "scooped",) ethical and legal issues, and technical or administrative issues. In many cases, motivation is less of an issue than are the perceived legal and technical issues in keeping an investigator from sharing their data. The perceived legal issues regarding privacy and confidentiality, and protecting the trust that the subject has when they give their time and effort to participate in a study, are what lead to multi-page DUAs.

Neuroimaging is not the only data type whose sharing is hampered by these privacy concerns. Genetic data is perhaps the most contentious to share; the eMERGE consortium worked through a number of issues with large-scale sharing of genetic data, including the usual administrative burdens and ethical concerns (McGuire et al., 2011), and the five sites of the consortium identified numerous inconsistencies across institutional policies due to concerns about ethical and legal protections. It is often easy to re-identify individuals from genetic data; one publication showing re-identification of individuals is even possible from pooled data (Homer et al., 2008), prompting the NIH to remove data from a public repository (Couzin, 2008). Despite the existence of more sophisticated re-identificationattacks (e.g., Schadt et al., 2012), the NIH has not responded by removing the data. One of the most recent efforts re-identified subjects through combining DNA sequences with publicly available, recreational genealogy databases (Gymrek et al., 2013). These publicized privacy breaches make patients rightly concerned about their identifiable health information being shared with unknown parties.

This leads to basically three categories of data that will never be made publicly available for easy access: (1) data that are non-shareable due to obvious re-identification concerns, such as extreme age of the subject or a zip code/disease combination that makes re-identification simple; (2) data that are non-shareable due to more complicated or less obvious concerns, such as genetic data or other data which may be re-identifiable in conjunction with other data not under the investigator's control; and (3) data that are non-shareable due to the local institutional review boards (IRBs) rules or other administrative decisions (e.g., stakeholders in the data collection not allowing sharing). For example, even with broad consent to share the data acquired at the time of data collection, some of the eMERGE sites were required to re-contact the subjects and re-consent prior to sharing within the eMERGE consortium, which can be a permanent show-stopper for some datasets (Ludman et al., 2010).

The first two data types may be shared with an appropriate DUA. But this does not guarantee "easy access;" it can slow down or even prevent research. This is particularly onerous when it is not known if the data being requested are actually useable for the particular analysis the data requestor is planning. For example, it may be impossible to tell how many subjects fit a particular set of criteria without getting access to the full data first (Vinterbo et al., 2012). It is markedly problematic to spend weeks, months, or even years waiting for access to a dataset, only to find out that of the several hundred subjects involved, only a few had usable combinations of data of sufficient quality necessary for one's analysis.

Problems with DUAs only become worse when trying to access data from multiple sites. Because each DUA is different, the administrative burden rapidly becomes unmanageable. In order to enable analyses across multiple sites, one successful approach is to share data derivatives. For example, the ENIGMA consortia pooled together data from many hundreds of local sites and thousands of subjects by providing analysis scripts to local sites and centrally collecting only the output of these scripts (Hilbar et al., 2013). Another example is DataSHIELD (Wolfson et al., 2010), which also uses shared summary measures to perform pooled analysis. These systems are good starting points, but they neither quantify privacy nor provide any guarantees against re-identification. In addition, summary measures are restricted to those that can be computed independently of other data. An analysis using ENIGMA cannot iterate among sites to compute results informed by the data as a whole. However, by allowing data holders to maintain control over access, such an approach does allow for more privacy protections at the cost of additional labor in implementing and updating a distributed architecture.

The ENIGMA approach is consistent with the *differential privacy* framework (Dwork et al., 2006), a strong notion of privacy which measures the risk of sharing the results of computations on private data. This quantification allows data holders to track overall risk, thereby allowing local sites to "opt-in" to analyses based on their own privacy concerns. However, in the differential privacy model, the computation is *randomized*—algorithms introduce noise to protect privacy, thereby making the computation less accurate. However, if protecting privacy permits sharing data derivatives, then aggregating private computations across many sites may lead to a benefit; even though each local computation is less accurate (to protect privacy), the "large N" benefit from many sites allowing access will still result in a more accurate computation.

The system we envision is a research consortium in which sites allow differentially-private computations on their data without requiring an individual DUA for each site. The data stays at each site, but the private data derivatives can be exchanged and aggregated to achieve better performance. In this paper we survey some of the relevant literature on differential privacy to clarify if and how it could help provide useful privacy protections in conjunction with distributed statistical analyses of neuroimaging data. The default situation is no data sharing: each site can only learn from its own data. We performed an experiment on neuroimages from a study to see if we could predict patients with schizophrenia from healthy control subjects. Protecting privacy permits a pooled analysis; without the privacy protections, each site would have to use its own data to learn a predictor. Our experiments show that by gathering differentially private classifiers learned from multiple sites, an aggregator can create a classifier that significant outperforms that which could be learned at a single site. This demonstrates the potential of differential privacy: sharing access to data derivatives (the classifiers) improves overall accuracy.

Many important research questions can be answered by the kind of large-scale neuroinformatics analyses that we envision.

- Regression is a fundamental statistical task. Regressing covariates such as age, diagnosis status, or response to a treatment against structure and function in certain brain regions (voxels in an image) is simple but can lead to important findings. For example, in examining the ability to aggregate structural imaging across different datasets (Fennema-Notestine et al., 2007) used the regression of age against brain volumes as a validity test. Age also affects resting state measures, as Allen et al. (2011) demonstrated on an aggregated dataset of 603 healthy subjects combined across multiple studies within an individual institution that had a commitment to data sharing and had minimal concerns regarding re-identification of the data. In that study, because privacy and confidentiality requirements that limited access to the full data, the logistics of extracting and organizing the data took the better part of a year (personal communication from the authors). In such a setting, asking a quick question such as whether age interacts with brain structure differently in healthy patients versus patients with a rare disorder would be impossible without submitting the project for IRB approval. This process can take months or even years and cost hundreds of dollars, whereas the analysis takes less than a day and may produce negative findings. We need a framework that facilitates access to data on the fly for such straightforward but fundamental analyses.

- The re-use of genetic data has been facilitated by dbGAP, NIH's repository for sharing genome-wide scan datasets, gene expression datasets, methylation datasets, and other genomic measures. The data need to be easily accessible for combined analysis for identification or confirmation of risk genes. The success of the Psychiatric Genomic Consortium in finding confirmed risk genes of schizophrenia after almost 5 years of aggregating datasets supports these goals of making every dataset re-usable (Ripke et al., 2013). While dbGAP has been a resounding success, it has its drawbacks. Finding the data can be a bit daunting, as often phenotype data is made available separately from the genetic data. For example, the PREDICT-HD Huntington's disease study rolled out a year before the genetic data. DbGAP's sharing requirements are driven by the need to ensure the data are handled appropriately and the subjects' confidentiality and privacy are protected; requesting a dataset entails both the PI and their institutional official signing an agreement as well as a review by the study designate. This process must be completed prior to access being granted or denied. As before, this precludes any exploratory analyses to identify particular needs, such as determining how many subjects have the all the required phenotype measures.
- The success of multimodal data integration in the analysis of brain structure/function (Plis et al., 2010; Bießmann et al., 2011; Bridwell et al., 2013; Schelenz et al., 2013), imaging/genetics (Liu et al., 2012; Chen et al., 2013; van Erp et al., 2013), and EEG/fMRI (Bridwell et al., 2013; Schelenz et al., 2013) shows that with enough data, we can go further than simple univariate linear models. For example, we can try to find combinations of features which predict the development of a disorder, response to various treatments, or relapse. With more limited data there has been some success in reproducing diagnostic classifications (Arbabshirani et al., 2013; Deshpande et al., 2013), and identifying coherent subgroupings within disorders which may have different genetic underpinnings (Girirajan et al., 2013). With combinations of imaging, genetic, and clinical profiles from thousands of subjects across autism, schizophrenia, and bipolar disorder, for example, we could aim to identify more clearly the areas of overlap and distinction, and what combinations of both static features and dynamic trajectories in the feature space identify clinically relevant clusters of subjects who may be symptomatically ambiguous.

2. PRIVACY MODELS AND DIFFERENTIAL PRIVACY

There are several different conceptual approaches to defining privacy in scenarios involving data sharing and computation. One approach is to create *de-identified* data; these methods take a database of records corresponding to individuals and create a *sanitized database* for use by the public or another party. Such approaches are used in official statistics and other settings—a survey of different privacy models can be found in Fung et al. (2010), and a survey of privacy technologies in a medical informatics context in Jiang et al. (2013). These approaches differ in how they define privacy and what guarantees they make with respect to this definition. For example, k-anonymity (Sweeney, 2002) quantifies privacy for a particular individual i with data x_i (for example, age and zip code) in terms of the number of other individuals whose data is also equal to x_i. Algorithms for guaranteeing k-anonymity manipulate data values (e.g., by reporting age ranges instead of exact ages) to enforce that each individual's record is identical to at least k other individuals.

A different conceptual approach to defining privacy is to try and quantify the change in the risk of re-identification as a result of publishing a function of the data. This differs from data sanitizing methods in two important respects. Firstly, privacy is a property of an algorithm operating on the data, rather an a property of the sanitized data—this is the difference between *semantic* and *syntactic* privacy. Secondly, it can be applied to systems which do not share data itself but instead share data derivatives (functions of the data). The recently proposed ϵ-differential privacy model (Dwork et al., 2006) quantifies privacy in terms of risk; it bounds the likelihood that someone can re-infer the data of an individual. Algorithms that guarantee differential privacy are *randomized*—they manipulate the data values (e.g., by adding noise) to bound the risk.

Finally, some authors define privacy in terms of data security and say that a data sharing system is private if it satisfies certain cryptographic properties. The most common of these models is secure multiparty computation (SMC) (Lindell and Pinkas, 2009), in which multiple parties can collaborate to compute a function of their data without leaking information about their private data to others. The guarantees are cryptographic in nature, and do not assess the re-inference or re-identification problem. For example, in a protocol to compute the maximum element across all parties, a successful execution would reveal the maximum. A secondary issue is developing practical systems to work on neuroinformatics data. Some progress has been made in this direction (Sadeghi et al., 2010; Huang et al., 2011; Nikolaenko et al., 2013), and it is conceivable that in a few years SMC will be implemented in real distributed systems.

2.1. PRIVACY TECHNOLOGIES FOR DATA SHARING

As discussed earlier, there are many scenarios in which sharing raw data is either difficult or impossible—strict DUAs, obvious re-identification issues, difficulties in assessing re-identifiability, and IRB or other policy rules. Similar privacy challenges exists in the secondary use of clinical data (National Research Council, 1997). In many medical research contexts, there has been a shift toward sharing *anonymized* data. The Health Insurance Portability and Accountability Act (HIPAA) privacy rule (45 CFR Part 160 and Subparts A and E of Part 164) allows the sharing of data as long as the data is de-identified. However, many approaches to anonymizing or "sanitizing" data sets (Sweeney, 2002; Li et al., 2007; Machanavajjhala et al., 2007; Xiao and Tao, 2007; Malin, 2008) are subject to attacks (Sweeney, 1997; Ganta et al., 2008; Narayanan and Shmatikov, 2008) that use public data to compromise privacy.

When data sharing itself is precluded, methods such as k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala et al., 2007), t-closeness (Li et al., 2007), and m-invariance (Xiao and

Tao, 2007) are no longer appropriate, since they deal with constructing private or sanitized versions of the data itself. In such situations we would want to construct data access *systems* in which data holders do not share the data itself but instead provide an interface to the data that allows certain pre-specified computations to be performed on that data. The data holder can then specify the granularity of access it is willing to grant subject to its policy constraints.

In this model of *interactive data access*, the software that controls the interface to the raw data acts as a "curator" that screens queries from outsiders. Each data holder can then specify the level of access which it will provide to outsiders. For example, a medical center may allow researchers to access summaries of clinical data for the purposes of exploratory analysis; a researcher can assess the feasibility of doing a study using existing records and then file a proposal with the IRB to access the real data (Murphy and Chueh, 2002; Murphy et al., 2006; Lowe et al., 2009; Vinterbo et al., 2012). In the neuroinformatics context, data holders may allow outside users to receive a histogram of average activity levels for regions of a certain size.

Being able to track the privacy risks in such an interactive system allows data holders to match access levels with local policy constraints. The key to privacy tracking is *quantification*—for each query or access to the data, a certain amount of information is "leaked" about the underlying data. With a sufficient number of queries it is theoretically possible to reconstruct the data (Dinur and Nissim, 2003), so the system should be designed to mitigate this threat and allow the data holders to "retire" data which has been accessed too many times.

2.2. DIFFERENTIAL PRIVACY

A user of the database containing private information may wish to apply a *query* or algorithm to the data. For example, they may wish to know the histogram of activity levels in a certain brain region for patients with a specified mutation. Because the answer to this query is of much lower dimension than a record in the database, it is tempting to regard disclosing the answer as not incurring a privacy risk. A important observation of Dinur and Nissim (2003) was that an adversary posing such queries may be able to reconstruct the entire database from the answers to multiple simple queries. The *differential privacy* model was introduced shortly thereafter, and has been adopted widely in the machine learning and data mining communities. The survey by Dwork and Smith (2009) covers much of the earlier theoretical work, and Sarwate and Chaudhuri (2013) review some works relevant to signal processing and machine learning. In the basic model, the database is modeled as a collection of N individuals' data records $\mathcal{D} = (x_1, x_2, \ldots, x_N)$, where x_j is the data for individual j. For example, x_j may be the MRI data associated to individual j together with information about mutations in certain genes for that individual.

An even simpler example is to estimate the mean activity in a certain region, so each x_j is simply a scalar which represented the measured activity of individual j. Let us call this desired algorithm Alg. Without any privacy constraint, the data curator would simply apply Alg to the data $\mathcal{D}$ to produce an output $h = \text{Alg}(\mathcal{D})$. However, in many cases the output h could compromise the privacy of the data and unfettered queries could lead to reidentification of an individual.

Under differential privacy, the curator applies an approximation PrivAlg to the data instead of Alg. The approximation PrivAlg is *randomized*—the randomness of the algorithm ensures that an observer of the output will have a difficult time re-identifying any individual in the database. More formally, PrivAlg$(\cdot)$ provides ϵ-differential privacy if for any subset of outputs $\mathcal{S}$,

$$\mathbb{P}\left(\text{PrivAlg}(\mathcal{D}) \in \mathcal{S}\right) \leq e^{\epsilon} \cdot \mathbb{P}\left(\text{PrivAlg}(\mathcal{D}') \in \mathcal{S}\right) \qquad (1)$$

for any two databases $\mathcal{D}$ and $\mathcal{D}'$ differing in a single individual. Here $\mathbb{P}(\cdot)$ is the probability over the randomness in the algorithm. It provides (ϵ, δ)-differential privacy if

$$\mathbb{P}\left(\text{PrivAlg}(\mathcal{D}) \in \mathcal{S}\right) \leq e^{\epsilon}\mathbb{P}\left(\text{PrivAlg}(\mathcal{D}') \in \mathcal{S}\right) + \delta. \qquad (2)$$

The guarantee that differential privacy makes is that the distribution of the output of PrivAlg does not change too much, regardless of whether any individual x_j is in the database or not. In particular, an adversary observing the output of PrivAlg and knowing all of the data of individuals in $\mathcal{D} \cap \mathcal{D}'$ common to both $\mathcal{D}$ and $\mathcal{D}'$ will still be uncertain of the remaining individual's data. Since this holds for any two databases which differ in one data point, each individual in the database is guaranteed of this protection. More specifically, the parameters ϵ and δ control the tradeoff between the false-alarm (Type I) and missed-detection (Type II) errors for an adversary trying to make a test between $\mathcal{D}$ and $\mathcal{D}'$ (see Oh and Viswanath, 2013 for a discussion).

Returning to our example of estimating the mean, the desired algorithm Alg is simply the sample mean of the m data points, so $\text{Alg}(\mathcal{D}) = \frac{1}{m}\sum_{j=1}^{m} x_i$. The algorithm Alg itself does not provide privacy because output is deterministic: the distribution of $\text{Alg}(\mathcal{D})$ is a point mass exactly at the average. If we change one data point to form, say $\mathcal{D}' = (x_1, x_2, \ldots, x_{m-1}, x'_m)$, then $\text{Alg}(\mathcal{D}') \neq \text{Alg}(\mathcal{D})$ and the only way Equation (1) can hold is if $\epsilon = \infty$. One form of a private algorithm is to add noise to the average (Dwork et al., 2006). A differentially private algorithm is $\text{PrivAlg}(\mathcal{D}) = \frac{1}{m}\sum_{j=1}^{m} x_i + \frac{1}{\epsilon m}z$, where z has a Laplace distribution with unit variance. The Laplace distribution is a popular choice, but there are many other distributions which can also guarantee differential privacy and may be better in some settings (Geng and Viswanath, 2012, 2013). For more general functions beyond averages, Gupte and Sundararajan (2010) and Ghosh et al. (2012) showed that in some cases we can find optimal mechanisms, while Nissim and Brenner (2010) show that this optimality may not be possible in general.

Although some variations on these basic definition have been proposed in the literature (Chaudhuri and Mishra, 2006; Rastogi et al., 2009; Kifer and Machanavajjhala, 2011), most of the literature focuses on ϵ- or (ϵ, δ)-differential privacy. Problems that have been studied in the literature range from statistical estimation (Smith, 2011; Kifer et al., 2012; Smith and Thakurta, 2013), to cover more complex data processing algorithms such as real-time signal processing (Fan and Xiong, 2012; Le Ny and Pappas, 2012a,b), classification (Chaudhuri et al., 2011; Rubinstein et al., 2012; Zhang et al., 2012b; Jain and Thakurta, 2014), online learning (Jain et al., 2012; Thakurta and Smith, 2013), dimensionality reduction (Hardt et al., 2012; Chaudhuri et al., 2013), graph estimation (Karwa et al., 2011; Kasiviswanathan et al., 2013), and auction design (Ghosh and Roth, 2011). The preceding citations are far from exhaustive, and new papers on differential privacy appear each month as methods and algorithms become more mature.

There are two properties of differential privacy which enable the kind of *privacy quantification* that we need in shared data-access scenarios. The first property is *post-processing invariance*: the output of an ϵ-differentially private algorithm PrivAlg maintains the same privacy guarantee—if $\hat{h} = \text{PrivAlg}(\mathcal{D})$, then the output of any function $g(\hat{h})$ applied to $\hat{h}$ is also ϵ-differentially private, provided $g(\cdot)$ doesn't depend on the data. This means that once the data curator has guaranteed ϵ-differential privacy for some computation, it need not track how the output is used in further processing. The second feature is *composition*—if we run two algorithms PrivAlg_1 and PrivAlg_2 on data $\mathcal{D}$ with privacy guarantees ϵ_1 and ϵ_2, then combined they have privacy risk at most $\epsilon_1 + \epsilon_2$. In some cases these composition guarantees can be improved (Dwork et al., 2010; Oh and Viswanath, 2013).

2.3. DIFFERENTIALLY PRIVATE ALGORITHMS

A central challenge in the use of differentially private algorithms is that by using randomization to protect privacy, the corresponding accuracy, or *utility*, of the result is diminished. We contend that the potential for a much larger sample size through data sharing makes this tradeoff worthwhile. In this section we discuss some of the differentially private methods for statistics and machine learning that have been developed in order to help balance privacy and utility in data analyses.

Differentially private algorithms have been developed for a number of important fundamental tasks in basic statistics and machine learning. Wasserman and Zhou (2010) put the differential privacy framework in a general statistical setting, and Smith (2011) studied point estimation, showing that many statistical quantities can be estimated with differential privacy with similar statistical efficiency. Duchi et al. (2012, 2013) studied a different version of *local* privacy and showed that requiring privacy essentially entails an increase in the sample size. Since differential privacy is related to the stability of estimators under changes in the data, Dwork and Lei (2009) and Lei (2011) used tools from robust statistics to design differentially private estimators. Williams and McSherry (2010) studied connections to probabilistic inference. More recently, Kifer et al. (2012) proposed methods for high-dimensional regression and Smith and Thakurta (2013) developed a novel variable selection method based on the LASSO.

One approach to designing estimators is the sample-and-aggregate (Nissim et al., 2007; Smith, 2011; Kifer et al., 2012), which uses subsampling of the data to build more robust estimators. This approach was applied to problems in sparse linear regression (Kifer et al., 2012), and in particular to analyze the LASSO (Smith and Thakurta, 2013) under the slightly weaker definition of (ϵ, δ)-differential privacy. There are several works which address convex optimization approaches to statistical model selection and machine learning under differential privacy (Chaudhuri et al., 2011; Kifer et al., 2012; Rubinstein et al., 2012; Zhang et al., 2012b) that encompass popular methods such as logistic regression, support vector machines, and other machine learning methods. Practical kernel-based methods for learning with differential privacy are still in their infancy (Chaudhuri et al., 2011; Jain and Thakurta, 2013).

2.4. CHALLENGES FOR DIFFERENTIAL PRIVACY

In addition to the theoretical and algorithmic developments, some authors have started trying to build end-to-end differentially private analysis toolkits and platforms. The query language PINQ (McSherry, 2010) was the first tool that allowed people to write differentially-private data-analysis programs that guarantee differential privacy, and has been used to write methods for a number of tasks, including network analyses (McSherry and Mahajan, 2010). Fuzz (Reed and Pierce, 2010) is a functional programming language that also guarantees differential privacy. At the systems level, AIRAVAT (Roy et al., 2010) is a differentially private version of MapReduce and GUPT (Mohan et al., 2012) uses the sample-and-aggregate framework to run general statistical algorithms such as k-means. One of the lessons from these implementations is that building a differentially private *system* involves keeping track of every data access—each access can leak some privacy—and systems can be vulnerable to attack from adversarial queries (Haeberlen et al., 2011).

A central challenge in designing differentially private algorithms for practical systems is setting the privacy risk level ϵ. In some cases, ϵ must be chosen to be quite large in order to produce useful results—such a case was studied in earlier work by Machanavajjhala et al. (2008) in the context of publishing differentially private statistics about commute times. On the other side, choosing a small value of ϵ may result in adding too much noise to allow useful analysis. To implement a real system, it is necessary to do a proper evaluation of the impact of ϵ on the utility of the results. Ultimately, the setting of ϵ is a policy decision that is informed by the privacy-utility tradeoff.

There are several difficulties with implementing existing methods "off the shelf" in the neuroinformatics context. Neuroimaging data is often continuous-valued. Much of the work on differential privacy has focused on discrete data, and algorithms for continuous data are still being investigated theoretically (Sarwate and Chaudhuri, 2013). In this paper we adapt existing algorithms, but there is a need to develop methods specifically designed for neuroimage analyses. In particular, images are high-dimensional signals, and differentially private version of algorithms such as PCA may perform poorly as the data dimension increases (Chaudhuri et al., 2013). Some methods do exist that exploit structural properties such as sparsity (Hardt and Roth, 2012, 2013), but there has been insufficient empirical investigation of these methods. Developing low-dimensional representations of the data (perhaps depending on the task) can help mitigate this.

Finally, neuroimaging datasets may contain few individuals. While the signal from each individual may be quite rich, the number of individuals in a single dataset may be small. Since privacy affects the statistical efficiency of estimators, we must develop distributed algorithms that can leverage the properties of datasets at many locations while protecting the privacy of the data at each. Small sample sizes present difficulties for statistical inference without privacy—the hope is that the larger sample size from sharing will improve statistical inference despite the impact of privacy considerations. We illustrate this in the next section.

3. APPLYING DIFFERENTIAL PRIVACY IN NEUROINFORMATICS

In the absence of a substitute for individual DUAs, sites are left to perform statistical analyses on their own data. Our proposal is to have sites participate in consortium in which they share differentially private data derivatives, removing the need for individual DUAs. Differential privacy worsens the quality of a statistical estimate at a single site because it introduces extra noise. However, because we can share the results of differentially private computations at different sites, we can reduce the impact of the noise from privacy. This larger effective sample size can give better estimates than are available at a single site, even with privacy. We illustrate this idea with two examples. The first is a simple problem of estimating the mean from noisy samples, and the second is an example of a classification problem.

3.1. ESTIMATING A MEAN

Perhaps the most fundamental statistical problem is estimating the mean of a variable. Suppose that we have N sites, each with m different samples of an unknown effect:

$$x_{i,j} = \mu + z_{i,j} \qquad i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, m, \tag{3}$$

where μ is an unknown mean, and $z_{i,j}$ is normally distributed noise with zero mean and unit variance. Each site can compute its local sample mean:

$$\bar{X}_i = \frac{1}{m}\sum_{j=1}^{m} x_{i,j} = \mu + \frac{1}{m}\sum_{j=1}^{m} z_{i,j}. \tag{4}$$

The sample mean $\bar{X}_i$ is a an estimate of μ which has an error that is normally distributed with zero mean and variance $\frac{1}{m}$. Thus a single site can estimate μ to within variance $\frac{1}{m}$. A simple ϵ-differentially private estimate of μ is

$$\tilde{X}_i = \frac{1}{m}\sum_{j=1}^{m} x_{i,j} + \frac{1}{\epsilon m} w_i, \tag{5}$$

where w_i is a Laplace random variable with unit variance. Thus a single site can make a differentially private estimate of μ with error variance $\frac{1}{m} + \frac{1}{(\epsilon m)^2}$. Now turning to the N sites, we can form an overall estimate using the differentially private local estimates:

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} \tilde{X}_i = \mu + \frac{1}{mN}\sum_{i=1}^{N}\sum_{j=1}^{m} x_{i,j} + \frac{1}{\epsilon m N}\sum_{i=1}^{N} w_i. \tag{6}$$

This is an estimate of μ with variance $\frac{1}{mN} + \frac{1}{(\epsilon m)^2 N}$.

The data sharing solution results in a lower error compared to the local non-private solution whenever $\frac{1}{m} > \frac{1}{mN} + \frac{1}{(\epsilon m)^2 N}$, or

$$N > 1 + \frac{1}{\epsilon^2 m}.$$

As the number of sites increases, we can support additional privacy at local nodes (ϵ can decrease) while achieving superior statistical performance over learning at a single site *without privacy*.

3.2. CLASSIFICATION

We now turn to a more complicated example of differentially private classification that shows how a public data set can be enhanced by information from differentially private analyses of additional data sets. In particular, suppose there are N sites with private data and 1 site with a publicly available dataset. Suppose private site i has m_i data points $\{(\vec{x}_{i,j}, y_{i,j}) : j = 1, 2, \ldots, m_i\}$, where each $\vec{x}_{i,j} \in \mathbb{R}^d$ is a d-dimensional vector of numbers representing features of the j-th individual at site i, and $y_{i,j} \in \{-1, 1\}$ is a label for that individual. For example, the data could be activity levels in certain voxels and the label could indicate a disease state. Each site can learn a classifier on its own local data by solving the following minimization problem.

$$\vec{w}_i = \underset{\vec{w}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{j=1}^{m_i} \ell(y_{i,j}\vec{w}^\top \vec{x}_{i,j}) + \frac{\lambda}{2}\|\vec{w}\|^2, \tag{7}$$

where $\ell(\cdot)$ is a loss function. This framework includes many popular algorithms: for the support vector machine (SVM) $\ell(z) = \max(0, 1 - z)$ and for logistic regression $\ell(z) = \log(1 + e^{-z})$.

Because the data at each site might be limited, they may benefit from producing differentially private versions $\vec{w}_i$ and then combining those with the public data to produce a better overall classifier. That is, leveraging many noisy classifiers may give better results than any $\vec{w}_i$ on its own. The method we propose is to train N differentially private classifiers using the objective perturbation method applied to the Huberized support vector machine (see Chaudhuri et al., 2011 for details). In this procedure, the local sites minimize a perturbed version of the classifier given in Equation (7). Let $\vec{w}_i$ be the differentially private classifier produced by site i.

Suppose the public data set has m_0 points $\{(\vec{x}_{0,j}, y_{0,j}) : j = 1, 2, \ldots, m_0\}$. We compute a new data set $\{(\vec{u}_{0,j}, y_{0,j}) : j = 1, 2, \ldots, m_0\}$ where $\vec{u}_{0,j}$ is an N-dimensional vector whose i-th component is equal to $\vec{w}_i^\top \vec{x}_{0,j}$. Thus $\vec{u}_{0,j}$ is the vector of "soft" predictions of the N differentially private classifiers produced by the private sites. The public site then uses logistic regression to train a new classifier:

$$\vec{w}_0 = \underset{\vec{w}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{j=1}^{m_0} \log(1 + e^{-y_{0,j}\vec{w}^\top u_{0,j}}) + \frac{\lambda}{2}\|\vec{w}\|^2. \tag{8}$$

This procedure is illustrated in **Figure 1**. The overall classification system produced by this procedure consists of the classifiers

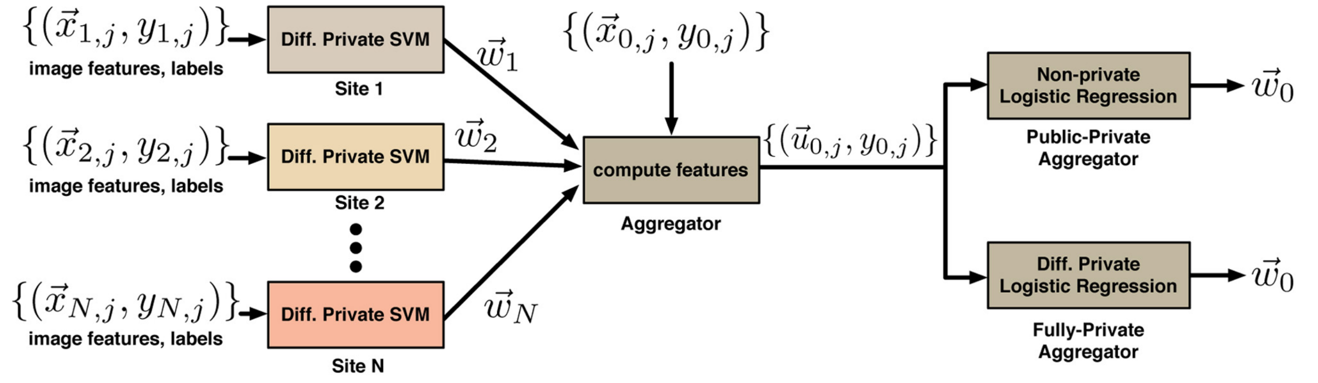


FIGURE 1 | System for differentially private classifier aggregation from many sites. The *N* sites each train a classifier on their local data to learn vectors $\{\vec{w}_i\}$. These are used by an aggregator to compute new features for its own data set. The aggregator can learn a classifier using its own data using a non-private algorithm (if its data is public) or a differentially private algorithm (if its data is private).

$\{\vec{w}_i : i = 0, 1, \ldots, N\}$. To classify a new point $\vec{x} \in \mathbb{R}^d$, the system computes $\vec{u} = (\vec{w}_1^\top \vec{x}, \vec{w}_2^\top \vec{x}, \ldots, \vec{w}_N^\top \vec{x})$ and then predicts the label $\hat{y} = \mathsf{sign}(\vec{w}_0^\top \vec{u})$. In the setting where the public site has more data, training a classifier on pairs $(\vec{u}, \vec{x})$ could also work better.

We can distinguish between two cases here—in the *public-private* case, described above, the classifier in Equation (8) uses differentially private classifiers from each of the *N* sites on public data, so the overall algorithm is differentially private with respect to the private data at the *N* sites. In the *fully-private* case, the data at the $(N + 1)$-th site is also private. In this case we can replace Equation (8) with a differentially private logistic regression method (Chaudhuri et al., 2011) to obtain a classifier which is differentially private with respect to the data at all $N + 1$ sites. Note, although we assign the role of constructing the overall two-level classifier to either the public-data site or one of the private sites in the real use-case no actual orchestrating of the process is required. It is convenient for the purposes of the demonstration (and without loss of generality) to treat a pre-selected site as an aggregator, which we do in the experiments below. **Figure 2**. can only be interpreted if we are consistent with the site that does the aggregation. However, all that needs to be done for the whole system to work is for the *N* (or $N + 1$ in the fully private case) private sites compute and publish their classifiers $\vec{w}_i$. Then in the public data case, anyone (even entities with no data), can construct and train a classifier by simply downloading the publicly available dataset and following the above-described procedure. This could be one of the sites with the private data as well. When no public data is available the second level classifier can be only computed by one of the private-data sites (or each one of them) and later published online to be useful even for entities with insufficient data. In both cases, the final classifier (or classifiers) is based on a larger data pool that is available to any single site.

From the perspective of differential privacy it is important to note that the only information that each site releases about its data is the separating hyperplane vector $\vec{w}_i$ and it does so only once. Considering privacy as a resource a site would want to minimize the loss of this resource. For that, a single release of information in our scheme is better that multiple exchanges in any of the iterative approaches (e.g., Gabay and Mercier, 1976; Zhang et al., 2012a).

We implemented the above system on a neuroimaging dataset (structural MRI scans) with $N = 10$ private sites. We combined data from four separate schizophrenia studies conducted at Johns Hopkins University (JHU), the Maryland Psychiatric Research Center (MPRC), the Institute of Psychiatry, London, UK (IOP), and the Western Psychiatric Institute and Clinic at the University of Pittsburgh (WPIC) (see Meda et al., 2008). The sample comprised 198 schizophrenia patients and 191 matched healthy controls (Meda et al., 2008). Our implementation relies on the differentially private SVM and logistic regression as described by Chaudhuri et al. (2011) and implementation available online[1]. The differentially private Hubertized SVM in our implementation used regularization parameter $\lambda = 0.01$, privacy parameter $\epsilon = 10$, and the Huber constant $h = 0.5$, while parameters for differentially private logistic regression were set to $\lambda = 0.01$ and $\epsilon = 10$ (for details see Chaudhuri et al., 2011). The quality of classification depends heavily on the quality of features; because distributed and differentially private feature learning algorithms are still under development, for the purposes of this example we assume features are given. To learn the features for this demonstration we used a restricted Boltzmann machine (RBM) (Hinton, 2000) with 50 sigmoidal hidden units. For training we have employed an implementation from Nitish Srivastava[2]. We have used L_1-regularization of the feature matrix $W(\lambda\|W\|_1)(\lambda = 0.1)$ and 50% dropout to encourage sparse features and effectively handle segmented gray matter images of 60465 voxels each. The learning rate parameter was set to 0.01. The weights were updated using the truncated Gibbs sampling method called contrastive divergence (CD) with a single sampling step (CD-1). Further information on RBM model can be found in Hinton (2000) and Hinton et al. (2006). After the RBM was trained we activated all 50 hidden units on each subject's MRI producing a 50 dimensional dataset. Note, no manual feature

[1] http://cseweb.ucsd.edu/~kamalika/code/dperm/

[2] https://github.com/nitishsrivastava/deepnet

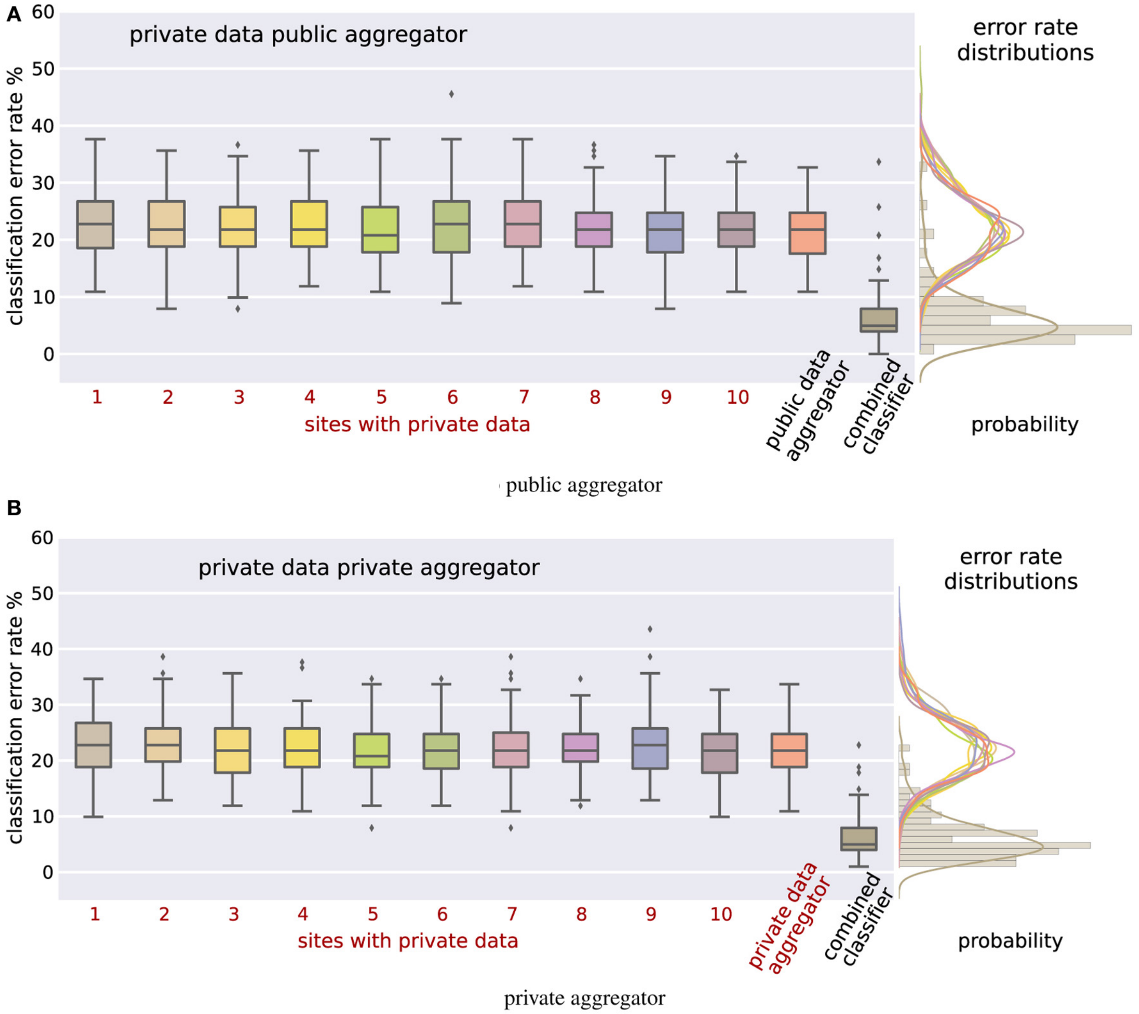


FIGURE 2 | Classification error rates for the mixed private-public case (A) and the fully-private case (B). In both cases the combined differentially private classifier performs significantly better than the individual classifiers. The difference is statistically significant even after Bonferroni correction (to account for multiple sites) with corrected *p*-values below 1.8×10^{-33}. Results thus motivate the use of differential privacy for sharing of brain imaging and genetic data to enable quick access to data which is either hard to access for logical reasons or not available for open sharing at all.

selection was involved as each and every feature was used. Using these features we repeated the following procedure 100 times:

1. Split the complete set of 389 subjects into class-balanced training and test sets comprising 70% (272 subjects) and 30% (117 subjects) of the data, respectively. The training set was split into $N + 1 = 11$ class-balanced subsets (sites) of 24 or 25 subjects each.
2. Train a differentially private SVM on $N = 10$ of these subsets independently (sites with private data).
3. Transform the data of the 11th subset (aggregator) using the trained SVM classifiers (as described above).
4. Train both a differentially private classifier (fully-private) and a standard logistic regression classifier (public-use) on the transformed dataset (combined classifier).
5. Compute the individual error rates on the test set for each of the $N = 10$ sites. Compute the error rates of a (differentially private) SVM trained on the data of 11th dataset and the aggregate classifier in Equation (8) that uses differentially private results from all of the sites.

The results that we obtained in this procedure are summarized in **Figure 2** for the mixed private-public (**Figure 2A**) as well as the fully-private (**Figure 2B**) cases. The 10 sites with private data all have base-line classification error rates of a little over 20%, indicating the relative difficulty of this classification task and highlighting the effect of the noise added for differential privacy. That is, on their own, each site would only be able to learn with that level of accuracy. The distribution of the error rates across experiments is given to the right. The last column of each figure shows the error rate of the combined classifier; **Figure 2A** shows the results for a public aggregator, and **Figure 2B** for the private aggregator. In both cases the error rate of the aggregated classifier is around 5%, which is a significant improvement over a single site. Additionally, the distribution of the error of the combined classifier is more tightly concentrated about its mean. To quantify the significance of the improvement we performed 2-sample t-tests for the distribution of the error rates of the combined classifier against error rate distributions of classifiers produced at individual sites. The largest Bonferroni corrected p-value was 1.8×10^{-33}. The experiments clearly show the benefits of sharing the results of differentially private computations over simply using the data at a single site. Even though the classifier that each site shares is a noisy version of what they could learn privately and thus less accurate, aggregating noisy classifiers produces at multiple sites dramatically lowers the resulting error.

4. DISCUSSION

Data sharing interfaces must take into account the realities of neuroimaging studies—current efforts have been very focused on the data structures and ability to query, retrieve and share complex and multi-modal datasets, usually under a fixed model of centralized warehousing, archiving, and privacy restrictions. There has been a remarkable lack of focus on the very important issues surrounding the lack of DUAs in older studies and also the privacy challenges which are growing as more data becomes available and predictive machine learning becomes more common.

We must consider several interlocking aspects when choosing a data sharing framework and the technology to enable it. Neuroimaging and genetics data present significant unique challenges for privacy. Firstly, this kind of data is very different from that considered by many works on privacy—images and sequence data are very high-dimensional and highly identifiable, which may set limits on what we expect to be achievable. Secondly, we must determine the data sharing structure—how is data being shared, and to whom. Institutional data holders may allow other institutions, individual researchers, or the public to access their data. The structure of the arrangement can inform which privacy technology is appropriate (Jiang et al., 2013). Thirdly, almost all privacy-preserving data sharing and data mining technologies are still under active research development and are not at the level of commercially deployed security technologies such as encryption for e-Commerce. A privacy-preserving computation model should be coupled with a legal and policy framework that allows enforcement in the case of privacy breaches. In our proposed model, sites can participate in a consortium in which only differentially private data derivatives are shared. By sharing access to the data, rather than the data itself, we mitigate the current proliferation of individually-generated DUAs, by allowing local data holders to maintain more control.

There are a number of challenges in building robust and scalable data sharing systems for neuroinformatics. On the policy side, standards and best practices should be established for data sharing within and across research consortia. For example, one major challenge is attribution and proper crediting for data used in large-scale studies. On the technology side, building federated data sharing systems requires additional fault-tolerance, security, and more sophisticated role-management than is typically found in the research environment. As noted by Haeberlen et al. (2011) implementing a differentially private system introduces additional security challenges without stricter access controls. Assigning different trust levels for different users (Vinterbo et al., 2012), managing privacy budgets, and other data governance policy issues can become quite complicated with differential privacy. On the statistical side, we must extend techniques from meta-analyses to interpret statistics computed from data sampled under heterogenous protocols. However, we believe these challenges can be overcome so that researchers can more effectively collaborate and learn from larger populations.

FUNDING

This work was supported by the National Institutes of Health via awards NIMH U01MH097435 (Lei Wang, PI) to Jessica A. Turner and NIBIB R01EB005846 and COBRE 5P20RR021938/P20GM103472 to Vince D. Calhoun.

ACKNOWLEDGMENTS

Anand D. Sarwate would like to acknowledge helpful discussions with K. Chaudhuri. All of the authors would like to thank the reviewers, whose extensive feedback helped to significantly improve the manuscript.

REFERENCES

Allen, E. A., Erhardt, E. B., Damaraju, E., Gruner, W., Segall, J. M., Silva, R. F., et al. (2011). A baseline for the multivariate comparison of resting state networks. *Front. Syst. Neurosci.* 5:2. doi: 10.3389/fnsys.2011.00002

Arbabshirani, M. R., Kiehl, K., Pearlson, G., and Calhoun, V. D. (2013). Classification of schizophrenia patients based on resting-state functional network connectivity. *Front. Neurosci.* 7:133. doi: 10.3389/fnins.2013.00133

Bießmann, F., Plis, S., Meinecke, F. C., Eichele, T., and Müller, K. R. (2011). Analysis of multimodal neuroimaging data. *IEEE Rev. Biomed. Eng.* 4, 6. doi: 10.1109/RBME.2011.2170675

Bridwell, D. A., Wu, L., Eichele, T., and Calhoun, V. D. (2013). The spatiospectral characterization of brain networks: fusing concurrent EEG spectra and fMRI maps. *Neuroimage* 69, 101–111. doi: 10.1016/j.neuroimage.2012.12.024

Chaudhuri, K., and Mishra, N. (2006). "When random sampling preserves privacy," in *Advances in Cryptology - CRYPTO 2006*. Lecture notes in computer science, Vol. 4117, ed C. Dwork (Berlin: Springer-Verlag), 198–213. doi: 10.1007/11818175_12

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011). Differentially private empirical risk minimization. *J. Mach. Learn. Res.* 12, 1069–1109.

Chaudhuri, K., Sarwate, A. D., and Sinha, K. (2013). A near-optimal algorithm for differentially-private principal components. *J. Mach. Learn. Res.* 14, 2905–2943.

Chen, J., Calhoun, V. D., Pearlson, G. D., Perrone-Bizzozero, N., Sui, J., Turner, J. A., et al. (2013). Guided exploration of genomic risk for gray matter abnormalities in schizophrenia using parallel independent component analysis with reference. *Neuroimage* 83, 384–396. doi: 10.1016/j.neuroimage.2013.05.073

Couzin, J. (2008). Genetic privacy. Whole-genome data not anonymous, challenging assumptions. *Science* 321, 1728. doi: 10.1126/science.321.5894.1278

Deshpande, G., Libero, L., Sreenivasan, K. R., Deshpande, H., and Kana, R. K. (2013). Identification of neural connectivity signatures of autism using machine learning. *Front. Hum. Neurosci.* 7:670. doi: 10.3389/fnhum.2013.00670

Dinur, I., and Nissim, K. (2003). "Revealing information while preserving privacy," in *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (New York, NY: ACM), 202–210.

Duchi, J., Jordan, M., and Wainwright, M. (2012). "Privacy aware learning," in *Advances in Neural Information Processing Systems 25*, eds P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (La Jolla, CA: Neural Information Processing Systems Foundation), 1439–1447.

Duchi, J., Wainwright, M. J., and Jordan, M. (2013). Local privacy and minimax bounds: sharp rates for probability estimation. *Adv. Neural Inform. Process. Syst.* 26, 1529–1537.

Dwork, C., and Lei, J. (2009). “Differential privacy and robust statistics,” in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC ’09)* (New York, NY: ACM), 371–380. doi: 10.1145/1536414.1536466

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). “Calibrating noise to sensitivity in private data analysis,” in *Theory of Cryptography*. Lecture notes in computer science, Vol. 3876, eds S. Halevi and T. Rabin (Berlin, Heidelberg: Springer), 265–284.

Dwork, C., Rothblum, G., and Vadhan, S. (2010). “Boosting and differential privacy,” in *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS ‘10)* (Las Vegas, NV), 51–60.

Dwork, C., and Smith, A. (2009). Differential privacy for statistics: what we know and what we want to learn. *J. Privacy Confident.* 1, 135–154.

Fan, L., and Xiong, L. (2012). “Real-time aggregate monitoring with differential privacy,” in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM ’12)* (New York, NY: ACM), 2169–2173.

Fennema-Notestine, C., Gamst, A. C., Quinn, B. T., Pacheco, J., Jernigan, T. L., Thal, L., et al. (2007). Feasibility of multi-site clinical structural neuroimaging studies of aging using legacy data. *Neuroinformatics* 5, 235–245. doi: 10.1007/s12021-007-9003-9

Fung, B. C. M., Wang, K., Chen, R., and Yu, P. S. (2010). Privacy-preserving data publishing: a survey of recent developments. *ACM Comput. Surv.* 42, 14:1–14:53. doi: 10.1201/9781420091502

Gabay, D., and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* 2, 17–40. doi: 10.1016/0898-1221(76)90003-1

Ganta, S. R., Kasiviswanathan, S. P., and Smith, A. (2008). “Composition attacks and auxiliary information in data privacy,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’08)* (New York, NY: ACM), 265–273. doi: 10.1145/1401890.1401926

Geng, Q., and Viswanath, P. (2012). *The Optimal Mechanism in Differential Privacy*. Technical Report arXiv:1212.1186.

Geng, Q., and Viswanath, P. (2013). *The Optimal Mechanism in (ϵ, δ)-Differential Privacy*. Technical Report arXiv:1305.1330.

Ghosh, A., and Roth, A. (2011). “Selling privacy at auction,” in *Proceeding of the 12th ACM Conference on Electronic Commerce (EC ’11)* (New York, NY: ACM), 199–208.

Ghosh, A., Roughgarden, T., and Sundararajan, M. (2012). Univerally utility-maximizing privacy mechanisms. *SIAM J. Comput.* 41, 1673–1693. doi: 10.1137/09076828X

Girirajan, S., Dennis, M. Y., Baker, C., Malig, M., Coe, B. P., Campbell, C. D., et al. (2013). Refinement and discovery of new hotspots of copy-number variation associated with autism spectrum disorder. *Am. J. Hum. Genet.* 92, 221–237. doi: 10.1016/j.ajhg.2012.12.016

Gupte, M., and Sundararajan, M. (2010). “Universally optimal privacy mechanisms for minimax agents,” in *ACM SIGMOD Symposium on Principles of Database Systems (PODS)* (New York, NY: ACM), 135–146.

Gymrek, M., McGuire, A. L., Folan, D., Halperin, E., and Erlich, Y. (2013). Identifying personal genomes by surname inference. *Science* 339, 321–324. doi: 10.1126/science.1229566

Haeberlen, A., Pierce, B. C., and Narayan, A. (2011). “Differential privacy under fire,” in *Proceedings of the 20th USENIX Conference on Security* (Berkeley, CA: USENIX Association).

Hardt, M., Ligett, K., and McSherry, F. (2012). “A simple and practical algorithm for differentially private data release,” in *Advances in Neural Information Processing Systems*, VOl. 25, eds P. Bartlett, F. Pereira, C. Burges, L. Bottou and K. Weinberger (La Jolla, CA: Neural Information Processing Systems Foundation), 2348–2356.

Hardt, M., and Roth, A. (2012). “Beating randomized response on incoherent matrices,” in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC ’12)* (New York, NY: ACM), 1255–1268.

Hardt, M., and Roth, A. (2013). “Beyond worst-case analysis in private singular vector computation,” in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC ’13)* (New York, NY: ACM), 331–340. doi: 10.1145/2488608.2488650

Hilbar, D., Calhoun, V., and Enigma Consortium (2013). “ENIGMA2: genome-wide scans of subcortical brain volumes in 16,125 subjects from 28 cohorts worldwide,” in *19th Annual Meeting of the Organization for Human Brain Mapping* (Seattle, WA).

Hinton, G. (2000). Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14, 2002. doi: 10.1162/089976602760128018

Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., et al. (2008). Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genet.* 4:e1000167. doi: 10.1371/journal.pgen.1000167

Huang, Y., Malka, L., Evans, D., and Katz, J. (2011). “Efficient privacy-preserving biometric identification,” in *Proceedings of the 18th Network and Distributed System Security Conference (NDSS 2011)*.

Jack, C. R., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., et al. (2008). The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods. *J. Magn. Reson. Imaging* 27, 685–691. doi: 10.1002/jmri.21049

Jain, P., Kothari, P., and Thakurta, A. (2012). “Differentially private online learning,” in *Proceedings of the 25th Annual Conference on Learning Theory (COLT ’12)*. JMLR workshop and conference proceedings, Vol. 23 (Scotland: Edinburgh), 24.1–24.34.

Jain, P., and Thakurta, A. (2013). “Differentially private learning with kernels,” in *Proceedings of The 30th International Conference on Machine Learning (ICML)*. JMLR: workshop and conference proceedings, Vol. 28, eds S. Dasgupta and D. McAllester (Beijing: International Machine Learning Society), 118–126.

Jain, P., and Thakurta, A. (2014). “(near) dimension independent risk bounds for differentially private learning,” in *Proceedings of the 31st International Conference on Machine Learning* (Atlanta, GA).

Jiang, X., Sarwate, A. D., and Ohno-Machado, L. (2013). Privacy technology to share data for comparative effectiveness research : a systematic review. *Med. Care* 51, S58–S65. doi: 10.1097/MLR.0b013e31829b1d10

Karwa, V., Raskhodnikova, S., Smith, A., and Yaroslavtsev, G. (2011). Private analysis of graph structure. *Proc. VLDB Endowment* 4, 1146–1157.

Kasiviswanathan, S., Nissim, K., Raskhodnikova, S., and Smith, A. (2013). “Analyzing graphs with node differential privacy,” in *Proceedings of the 10th Theory of Cryptography Conference (TCC)* (Tokyo), 457–476. doi: 10.1007/978-3-642-36594-2_26

Kifer, D., and Machanavajjhala, A. (2011). “No free lunch in data privacy,” in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (New York, NY: ACM), 193–204. doi: 10.1145/1989323.1989345

Kifer, D., Smith, A., and Thakurta, A. (2012). “Private convex empirical risk minimization and high-dimensional regression,” in *Proceedings of the 25th Annual Conference on Learning Theory (COLT ’12). JMLR Workshop and Conference Proceedings*, Vol. 23, eds S. Mannor, N. Srebro and R. C. Williamson (Scotland: Edinburgh), 25.1–25.40.

Lei, J. (2011). “Differentially private M-estimators,” in *Advances in Neural Information Processing Systems 24*, eds J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. Pereira and K. Q. Weinberger (La Jolla, CA: Neural Information Processing Systems Foundation), 361–369.

Le Ny, J., and Pappas, G. J. (2012a). “Differentially private filtering,” in *Proceedings of the 51st Conference on Decision and Control (CDC)* (Maui, HI), 3398–3403.

Le Ny, J., and Pappas, G. J. (2012b). “Differentially private Kalman filtering,” in *Proceedings of the 50th Annual Allerton Conference on Communications, Control and Computing* (Monticello, IL), 1618–1625.

Lindell, Y., and Pinkas, B. (2009). Secure multiparty computation for privacy-preserving data mining. *J. Priv. Confident.* 1, 59–98.

Li, N., Li, T., and Venkatasubramanian, S. (2007). “t-closeness: privacy beyond k-anonymity and ℓ-diversity,” in *IEEE 23rd International Conference on Data Engineering (ICDE)* (Istanbul), 106–115.

Liu, J., Ghassemi, M. M., Michael, A. M., Boutte, D., Wells, W., Perrone-Bizzozero, N., et al. (2012). An ica with reference approach in identification of genetic variation and associated brain networks. *Front. Hum. Neurosci.* 6:21. doi: 10.3389/fnhum.2012.00021

Lowe, H. J., Ferris, T. A., Hernandez, P. M., and Weber, S. C. (2009). “Stride–an integrated standards-based translational research informatics platform,” in *Proceedings of the 2009 AMIA Annual Symposium* (San Francisco, CA), 391–395.

Ludman, E. J., Fullerton, S. M., Spangler, L., Trinidad, S. B., Fujii, M. M., Jarvik, G. P., et al. (2010). Glad you asked: participants' opinions of re-consent for dbGaP data submission. *J. Empir. Res. Hum. Res. Ethics* 5, 9–16. doi: 10.1525/jer.2010.5.3.9

Machanavajjhala, A., Kifer, D., Abowd, J. M., Gehrke, J., and Vilhuber, L. (2008). "Privacy: theory meets practice on the map," in *IEEE 24th International Conference on Data Engineering (ICDE)* (Cancun), 277–286.

Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). *l*-diversity: privacy beyond *k*-anonymity. *ACM Trans. Knowl. Discov. Data* 1, 3. doi: 10.1145/1217299.1217302

Malin, B. (2008). k-unlinkability: a privacy protection model for distributed data. *Data Knowl. Eng.* 64, 294–311. doi: 10.1016/j.datak.2007.06.016

McGuire, A. L., Basford, M., Dressler, L. G., Fullerton, S. M., Koenig, B. A., Li, R., et al. (2011). Ethical and practical challenges of sharing data from genome-wide association studies: the eMERGE consortium experience. *Genome Res.* 21, 1001–1007. doi: 10.1101/gr.120329.111

McSherry, F. (2010). Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Commun. ACM* 53, 89–97. doi: 10.1145/1810891.1810916

McSherry, F., and Mahajan, R. (2010). "Differentially-private network trace analysis," in *Proceedings of SIGCOMM* (New Delhi).

Meda, S. A., Giuliani, N. R., Calhoun, V. D., Jagannathan, K., Schretlen, D. J., Pulver, A., et al. (2008). A large scale (*n*= 400) investigation of gray matter differences in schizophrenia using optimized voxel-based morphometry. *Schizophr. Res.* 101, 95–105. doi: 10.1016/j.schres.2008.02.007

Mennes, M., Biswal, B. B., Castellanos, F. X., and Milham, M. P. (2013). Making data sharing work: the fcp/indi experience. *Neuroimage* 82, 683–691. doi: 10.1016/j.neuroimage.2012.10.064

Mohan, P., Thakurta, A., Shi, E., Song, D., and Culler, D. (2012). "GUPT: privacy preserving data analysis made easy," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (New York, NY: ACM), 349–360. doi: 10.1145/2213836.2213876

Murphy, S. N., and Chueh, H. (2002). "A security architecture for query tools used to access large biomedical databases," in *AMIA, Fall Symposium 2002*, 552–556.

Murphy, S. N., Mendis, M. E., Berkowitz, D. A., Kohane, I., and Chueh, H. C. (2006). "Integration of clinical and genetic data in the i2b2 architecture," in *Proceedings of the 2006 AMIA Annual Symposium* (Washington, DC), 1040.

Narayanan, A., and Shmatikov, V. (2008). "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (Oakland, CA), 111–125. doi: 10.1109/SP.2008.33

National Research Council. (1997). *The Computer-Based Patient Record: An Essential Technology for Health Care, Revised Edition*. Washington, DC: The National Academies Press.

Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., and Taft, N. (2013). "Privacy-preserving ridge regression on hundreds of millions of records," in *IEEE Symposium on Security and Privacy* (San Francisco, CA), 334–348.

Nissim, K., and Brenner, H. (2010). "Impossibility of differentially private universally optimal mechanisms," in *IEEE Symposium on Foundations of Computer Science (FOCS)* (Las Vegas, NV).

Nissim, K., Raskhodnikova, S., and Smith, A. (2007). "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing (STOC '07)* (New York, NY: ACM), 75–84. doi: 10.1145/1250790.1250803

Oh, S., and Viswanath, P. (2013). *The composition theorem for differential privacy*. Technical Report arXiv:1311.0776 [cs.DS].

Plis, S. M., Calhoun, V. D., Eichele, T., Weisend, M. P., and Lane, T. (2010). MEG and fMRI fusion for nonlinear estimation of neural and BOLD signal changes. *Front. Neuroinform.* 4:12. doi: 10.3389/fninf.2010.00114

Poldrack, R. A., Barch, D. M., Mitchell, J. P., Wager, T. D., Wagner, A. D., Devlin, J. T., et al. (2013). Toward open sharing of task-based fMRI data: the OpenfMRI project. *Front. Neuroinform.* 7:12. doi: 10.3389/fninf.2013.00012

Poline, J.-B., Breeze, J. L., Ghosh, S., Gorgolewski, K., Halchenko, Y. O., Hanke, M., et al. (2012). Data sharing in neuroimaging research. *Front. Neuroinform.* 6:9. doi: 10.3389/fninf.2012.00009

Potkin, S. G., and Ford, J. M. (2009). Widespread cortical dysfunction in schizophrenia: the FBIRN imaging consortium. *Schizophr. Bull.* 35, 15–18. doi: 10.1093/schbul/sbn159

Rastogi, V., Hay, M., Miklau, G., and Suciu, D. (2009). "Relationship privacy: output perturbation for queries with joins," in *Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '09)* (New York, NY: ACM), 107–116. doi: 10.1145/1559795.1559812

Reed, J., and Pierce, B. C. (2010). "Distance makes the types grow stronger: a calculus for differential privacy," in *ACM SIGPLAN International Conference on Functional Programming (ICFP)* (Baltimore, MD).

Ripke, S., O'Dushlaine, C., Chambert, K., Moran, J. L., Kahler, A. K., Akterin, S., et al. (2013). Genome-wide association analysis identifies 13 new risk loci for schizophrenia. *Nat. Genet.* 45, 1150–1159. doi: 10.1038/ng.2742

Roy, I., Setty, S. T. V., Kilzer, A., Shmatikov, V., and Witchel., E. (2010). "Airavat: security and privacy for MapReduce," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI '10)* (Berkeley, CA: USENIX Association).

Rubinstein, B. I. P., Bartlett, P. L., Huang, L., and Taft, N. (2012). Learning in a large function space: privacy-preserving mechanisms for SVM learning. *J. Priv. Confident.* 4, 65–100.

Sadeghi, A.-R., Schneider, T., and Wehrenberg, I. (2010). "Efficient privacy-preserving face recognition," in *Information, Security and Cryptology – ICISC 2009*. Lecture notes in computer science, Vol. 5984, eds D. Lee and S. Hong (Berlin: Springer), 229–244. doi: 10.1007/978-3-642-14423-3_16

Sarwate, A. D., and Chaudhuri, K. (2013). Signal processing and machine learning with differential privacy: theory, algorithms, and challenges. *IEEE Signal Process. Mag.* 30, 86–94. doi: 10.1109/MSP.2013.2259911

Schadt, E. E., Woo, S., and Hao, K. (2012). Bayesian method to predict individual SNP genotypes from gene expression data. *Nat. Genet.* 44, 603–608. doi: 10.1038/ng.2248

Schelenz, P. D., Klasen, M., Reese, B., Regenbogen, C., Wolf, D., Kato, Y., et al. (2013). Multisensory integration of dynamic emotional faces and voices: method for simultaneous EEG-fMRI measurements. *Front. Hum. Neurosci.* 7:729. doi: 10.3389/fnhum.2013.00729

Smith, A. (2011). "Privacy-preserving statistical estimation with optimal convergence rates," in *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC '11)* (New York, NY: ACM), 813–822.

Smith, A., and Thakurta, A. (2013). "Differentially private feature selection via stability arguments, and the robustness of LASSO," in *Conference on Learning Theory*. JMLR: workshop and conference proceedings, Vol. 30, 1–32.

Sweeney, L. (1997). Weaving technology and policy together to maintain confidentiality. *J. Law Med. Ethics* 25, 98–110. doi: 10.1111/j.1748-720X.1997.tb01885.x

Sweeney, L. (2002). k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzz. Knowl. Based Syst.* 10, 557–570. doi: 10.1142/S0218488502001648

Thakurta, A. G., and Smith, A. (2013). "(Nearly) optimal algorithms for private online learning in full-information and bandit settings," in *Advances in Neural Information Processing Systems 26*, eds C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, 2733–2741.

Turner, J. A., and Van Horn, J. D. (2012). Electronic data capture, representation, and applications for neuroimaging. *Front. Neuroinform.* 6:16. doi: 10.3389/fninf.2012.00016

van Erp, T. G., Guella, I., Vawter, M. P., Turner, J., Brown, G. G., McCarthy, G., et al. (2013). Schizophrenia miR-137 locus risk genotype is associated with dorsolateral prefrontal cortex hyperactivation. *Biol. Psychiatry* 75, 398–405. doi: 10.1016/j.biopsych.2013.06.016

Vinterbo, S. A., Sarwate, A. D., and Boxwala, A. (2012). Protecting count queries in study design. *J. Am. Med. Inform. Assoc.* 19, 750–757. doi: 10.1136/amiajnl-2011-000459

Wasserman, L., and Zhou, S. (2010). A statistical framework for differential privacy. *J. Am. Stat. Assoc.* 105, 375–389. doi: 10.1198/jasa.2009.tm08651

Williams, O., and McSherry, F. (2010). "Probabilistic inference and differential privacy," in *Advances in Neural Information Processing Systems 23*, eds J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, 2451–2459.

Wolfson, M., Wallace, S., Masca, N., Rowe, G., Sheehan, N., Ferretti, V., et al. (2010). DataSHIELD: resolving a conflict in contemporary bioscience - performing a pooled analysis of individual-level data without sharing the data. *Int. J. Epidemiol.* 39, 1372–1382. doi: 10.1093/ije/dyq111

Xiao, X., and Tao, Y. (2007). "*m*-invariance: towards privacy preserving re-publication of dynamic datasets," in *Proceedings of the 2007 ACM SIGMOD*

International Conference on Management of Data (New York, NY: ACM), 689–700. doi: 10.1145/1247480.1247556

Zhang, C., Lee, H., and Shin, K. G. (2012a). "Efficient distributed linear classification algorithms via the alternating direction method of multipliers," in *International Conference on Artificial Intelligence and Statistics* (La Palma), 1398–1406.

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., and Winslett, M. (2012b). Functional mechanism: regression analysis under differential privacy. *Proc. VLDB Endowment* 5, 1364–1375.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 18 December 2013; paper pending published: 20 February 2014; accepted: 19 March 2014; published online: 07 April 2014.

Citation: Sarwate AD, Plis SM, Turner JA, Arbabshirani MR and Calhoun VD (2014) Sharing privacy-sensitive access to neuroimaging and genetics data: a review and preliminary validation. Front. Neuroinform. ***8****:35. doi: 10.3389/fninf.2014.00035*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 11 March 2014
doi: 10.3389/fninf.2014.00021

Pain Research Forum: application of scientific social media frameworks in neuroscience

Sudeshna Das[1,2]*†, Patricia G. McCaffrey[3]†, Megan W. T. Talkington[3], Neil A. Andrews[3], Stéphane Corlosquet[1], Adrian J. Ivinson[3] and Tim Clark[1,2,4]

[1] MassGeneral Institute for Neurodegenerative Disease, Massachusetts General Hospital, Cambridge, MA, USA
[2] Department of Neurology, Harvard Medical School, Boston, MA, USA
[3] Harvard NeuroDiscovery Center, Harvard Medical School, Boston, MA, USA
[4] School of Computer Science, University of Manchester, Manchester, UK

Edited by:
Xi Cheng, Lieber Institue for Brain Development, USA

Reviewed by:
Dimitrios A. Adamos, Aristotle University of Thessaloniki, Greece
Rochelle Ackerley, University of Gothenburg, Sweden
Syed Anwar, University of Engineering and Technology, Pakistan

****Correspondence:***
Sudeshna Das, MassGeneral Institute for Neurodegenerative Disease, Massachusetts General Hospital, 65 Landsdowne St., Cambridge, MA 02139, USA
e-mail: sdas5@mgh.harvard.edu

†These authors have contributed equally to this work.

Background: Social media has the potential to accelerate the pace of biomedical research through online collaboration, discussions, and faster sharing of information. Focused web-based scientific social collaboratories such as the Alzheimer Research Forum have been successful in engaging scientists in open discussions of the latest research and identifying gaps in knowledge. However, until recently, tools to rapidly create such communities and provide high-bandwidth information exchange between collaboratories in related fields did not exist.

Methods: We have addressed this need by constructing a reusable framework to build online biomedical communities, based on Drupal, an open-source content management system. The framework incorporates elements of Semantic Web technology combined with social media. Here we present, as an exemplar of a web community built on our framework, the Pain Research Forum (PRF) (http://painresearchforum.org). PRF is a community of chronic pain researchers, established with the goal of fostering collaboration and communication among pain researchers.

Results: Launched in 2011, PRF has over 1300 registered members with permission to submit content. It currently hosts over 150 topical news articles on research; more than 30 active or archived forum discussions and journal club features; a webinar series; an editor-curated weekly updated listing of relevant papers; and several other resources for the pain research community. All content is licensed for reuse under a Creative Commons license; the software is freely available. The framework was reused to develop other sites, notably the Multiple Sclerosis Discovery Forum (http://msdiscovery.org) and StemBook (http://stembook.org).

Discussion: Web-based collaboratories are a crucial integrative tool supporting rapid information transmission and translation in several important research areas. In this article, we discuss the success factors, lessons learned, and ongoing challenges in using PRF as a driving force to develop tools for online collaboration in neuroscience. We also indicate ways these tools can be applied to other areas and uses.

Keywords: social media, neuropathic pain, content management systems, Drupal

INTRODUCTION

Biomedical scientists rely heavily on the World Wide Web and Internet to do research and to perform literature, database, and information searches. Researchers are also increasingly adopting the Web to collaborate and exchange ideas. Web-based communities that bring together scientists from different disciplines, institutions, and sectors are called "collaboratories" (National Research Council, 1993). Collaboratories with embedded social media tools can increase the pace and quality of scientific collaboration with rapid, open and structured communication (Kouzes et al., 1996; Finholt and Olson, 1997).

However, several challenges for effective collaboration exist with respect to trust, independence, attribution, and intellectual property (Bos et al., 2007; Clark and Kinoshita, 2007). Scientists may prefer to work independently and may be restricted by the boundaries of institutions to freely exchange ideas. Moreover, there is the added complexity of communications across disciplines. Alzforum (http://www.alzforum.org) (Kinoshita and Clark, 2007)—a community of Alzheimer's disease researchers—was successfully able to overcome these challenges with a combination of high quality articles, neutrality, inclusiveness and editorial solicitation/moderation to gain trust and participation (Clark and Kinoshita, 2007). The Schizophrenia Research Forum (SRF) (http://www.schizophreniaforum.org/) was modeled after Alzforum to focus on schizophrenia research. SRF was built using the same software code as the original Alzforum site, and thus has an identical look, feel, and functionality to the previous version of Alzforum, before its 2013 re-launch. However, tools to rapidly create and customize such communities were not readily available and therefore the software cost to launch one such community could not be effectively amortized across others using a common software model.

To address this need, we decided to create a reusable platform to build biomedical web communities. We developed the Science Collaboration Framework (SCF) to provide the building blocks for these communities. An earlier version of the platform was developed primarily to publish scholarly articles in biomedicine that can be indexed by the National Library of Medicine (NLM) PubMed library (Das et al., 2009). Our newer version has been re-engineered to add social media and community features similar to those available in Alzforum. We also incorporated elements of Semantic Web technologies (Berners-Lee et al., 2001) to facilitate interoperability and interdisciplinary communications. Semantic Web is a technology developed by the World Wide Web Consortium (W3C), which aims to promote interoperability of Web content and creation of a "Web of Data" through the use of machine-readable Web pages. It relies heavily upon the used of agreed common vocabularies to describe objects and relationships in the Web. Biomedical science has developed a very rich set of such vocabularies or "ontologies" (over 300 vocabularies with over 5 million terms registered in the National Center for Biomedical Ontology at Stanford University Medical School). Use of ontologies such as these permits, among other benefits, resolution of the many synonym terms in biomedicine, to single common identifiers. Enabling Semantic Web technologies on our framework is meant as a step toward better integration with biomedical vocabularies and databases.

We chose the research area of chronic pain as the first use case. Chronic pain significantly impacts quality of life and is a substantial, growing, and unmet medical need worldwide. Although researchers have made great strides in understanding the underlying mechanisms and neurobiology of pain, few of these discoveries have been translated into new treatments. According to a recent report from the US Institutes of Medicine, chronic pain affects an estimated 100 million people in the US, and costs $600 billion annually in health care and lost productivity (National Research Council, 2011); the world-wide toll is unknown. For the most part, chronic pain conditions lack medications that are effective and well tolerated. One of the roadblocks to new treatments is a lack of communication and collaboration between basic, translational, and clinical researchers in the diverse scientific fields and clinical specialties that make up the pain research community. Thus, we developed an online open community, Pain Research Forum (PRF, http://painresearchforum.org), for pain researchers to freely exchange ideas and collectively elevate discussion of the causes of chronic pain and how that knowledge can be translated into new treatments and better care.

MATERIALS AND METHODS

We have developed a reusable platform—SCF—to build science communities in focused biomedical areas. Previous versions of the platform included tools to publish scientific review articles following the NLM Document Type Definition (DTD), which can be indexed in PubMed (Das et al., 2009). The new release, used for PRF, includes a large number of additional community features, including means to publish news articles, forums, member profiles and various community and research resources. These features are described in the following sections.

ARCHITECTURE

The SCF is developed on an open-source content management system, Drupal 7[1]. Drupal is based on the PHP programming language and MySQL database running in the Linux/Apache web server environment. We chose Drupal because it is easily extensible and there are 30,000 registered Drupal developers continually contributing modules and enhancements to the core features[2]. We developed several custom content types and packaged them as features that can be installed and reused on any science community site. The graphic design (colors, fonts, etc.) is customized for each site using a theme layer (Kumar, 2012). The key content types available in the SCF are described in the next sections. The software is freely available upon request and a complete manual for editors to manage the site is under development.

NEWS ARTICLES AND FORUMS

The number of papers published in scientific journals continues to grow at a double-exponential rate (Hunter and Cohen, 2006) and it is becoming increasingly difficult for researchers to keep up with the literature. One way to address this problem is to publish news articles that summarize the research and provide context. We have developed a news feature that allows editors of the site to readily publish original news articles on emerging research. The News article has the following main fields: title, subhead, author, and body. The body text is composed in a WYSIWYG editor that allows flexible styling and the ability to add images. News items have references that are implemented as links to bibliographic listings of papers (described in section Papers of the Week). We wanted an easy method for researchers to find relevant news from a certain field, thus news items are categorized with terms from a pre-defined taxonomy. News articles can also be related to other news stories or papers, which appear in a block to the right of the article.

Discussion forums are important social media tools that enable interactions among researchers. Forums have fields similar to those of News. Forums can be Discussions of open research questions, structured Webinars or Journal Clubs. Videos and images can be embedded in Discussion, Webinars, or Journal Clubs. Site editors can specify related items for any Discussion, Webinar, or Journal Club.

Each News article and Forum can be commented on, bookmarked, watched, recommended or shared using social media tools.

PAPERS OF THE WEEK

Hundreds of papers are published weekly in PubMed for a given biomedical domain such as pain, and editor-curated weekly digests can help researchers stay abreast of the growing literature.

[1] http://www.drupal.org

[2] http://en.wikipedia.org/wiki/Drupal

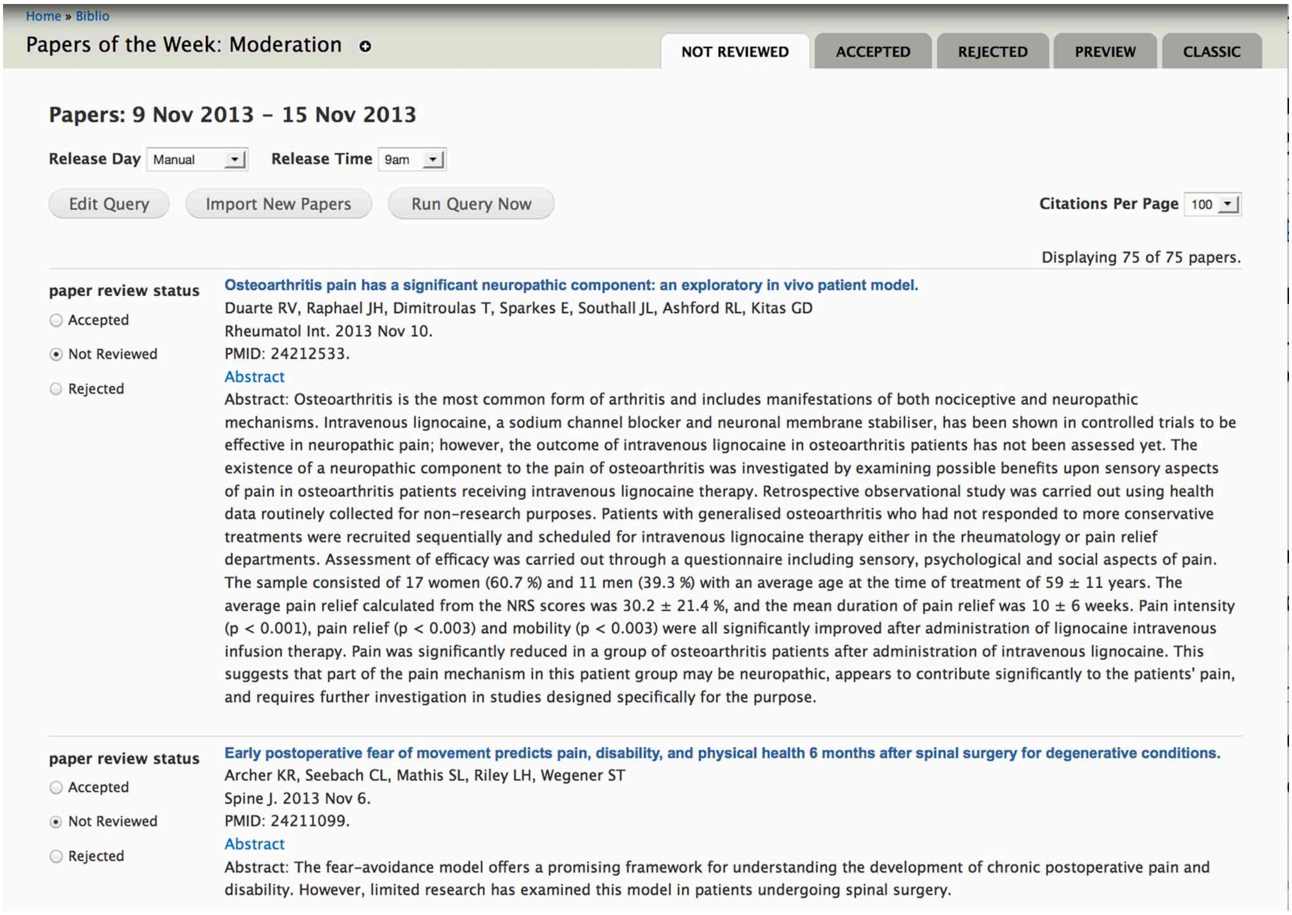
Thus, we decided to create a module for curating and annotating papers downloaded from the NLM PubMed library. We use the previously developed PubMed module to import biomedical articles from PubMed using its Application Programming Interface (API) (Sayers, 2008). The Drupal biblio module[3] is used to represent papers. We further developed the Journal Stream module that runs nightly queries using the software utility cron and imports the results in batches of 100 items. Complete documentation including a screencast and software for the Journal Stream module is available[4].

Each imported item is presented to the editors in a moderation queue and can then be "accepted" or "rejected." A screen shot of the moderation queue is shown in **Figure 1**. The abstract of each paper is displayed so that editors can determine whether the paper should be accepted or rejected. Key papers can be selected as Editors' Picks, and editors can choose to highlight the paper with a few sentences. The accepted papers are published as weekly collections; the periodicity of posting the collections can be configured per site. Once the collection is published, each paper can be individually commented on, bookmarked, watched, recommended or shared using social media tools. Users may download paper citations into the EndNote reference management software using the Endnote XML format. Currently, only EndNote is available to PRF users, but the Drupal biblio module allows site administrators to enable the BibTeX format if desired, for import into various other reference management tools such as Mendeley, Reference Manager, or Papers.

The citation and Medline Subject Headings (MeSH) terms associated with each paper are automatically updated periodically, as new information is posted in PubMed.

MEMBERS AND REGISTRATION

Members are the most important component of an online scientific community. We developed tools for members to join the site and publish their profile. While much of the content on the site is freely accessible without registration, only members can post comments on the site and have access to other members' profiles. The registration process starts with the members signing up online using a form. Research credentials including affiliation, position and research interest fields are required. The full name, email, city, and country are also required. Members agree to terms and conditions of membership on the site. The editor is



Papers of the Week: Moderation

Papers: 9 Nov 2013 – 15 Nov 2013

Release Day Manual

Release Time 9am

Citations Per Page 100

Displaying 75 of 75 papers.

paper review status

○ Accepted

● Not Reviewed

○ Rejected

Osteoarthritis pain has a significant neuropathic component: an exploratory in vivo patient model.

Duarte RV, Raphael JH, Dimitroulas T, Sparkes E, Southall JL, Ashford RL, Kitas GD

Rheumatol Int. 2013 Nov 10.

PMID: 24212533.

Abstract

Abstract: Osteoarthritis is the most common form of arthritis and includes manifestations of both nociceptive and neuropathic mechanisms. Intravenous lignocaine, a sodium channel blocker and neuronal membrane stabiliser, has been shown in controlled trials to be effective in neuropathic pain; however, the outcome of intravenous lignocaine in osteoarthritis patients has not been assessed yet. The existence of a neuropathic component to the pain of osteoarthritis was investigated by examining possible benefits upon sensory aspects of pain in osteoarthritis patients receiving intravenous lignocaine therapy. Retrospective observational study was carried out using health data routinely collected for non-research purposes. Patients with generalised osteoarthritis who had not responded to more conservative treatments were recruited sequentially and scheduled for intravenous lignocaine therapy either in the rheumatology or pain relief departments. Assessment of efficacy was carried out through a questionnaire including sensory, psychological and social aspects of pain. The sample consisted of 17 women (60.7 %) and 11 men (39.3 %) with an average age at the time of treatment of 59 ± 11 years. The average pain relief calculated from the NRS scores was 30.2 ± 21.4 %, and the mean duration of pain relief was 10 ± 6 weeks. Pain intensity ($p < 0.001$), pain relief ($p < 0.003$) and mobility ($p < 0.003$) were all significantly improved after administration of lignocaine intravenous infusion therapy. Pain was significantly reduced in a group of osteoarthritis patients after administration of intravenous lignocaine. This suggests that part of the pain mechanism in this patient group may be neuropathic, appears to contribute significantly to the patients' pain, and requires further investigation in studies designed specifically for the purpose.

paper review status

○ Accepted

● Not Reviewed

○ Rejected

Early postoperative fear of movement predicts pain, disability, and physical health 6 months after spinal surgery for degenerative conditions.

Archer KR, Seebach CL, Mathis SL, Riley LH, Wegener ST

Spine J. 2013 Nov 6.

PMID: 24211099.

Abstract

Abstract: The fear-avoidance model offers a promising framework for understanding the development of chronic postoperative pain and disability. However, limited research has examined this model in patients undergoing spinal surgery.

FIGURE 1 | Papers of the Week moderation queue. Papers are imported nightly from NLM PubMed using a tailored query. Editors are presented with an easy-to-use interface to accept or reject the papers.

[3] https://drupal.org/project/biblio

[4] http://scf.github.io/journalstream/

notified when a member signs up and once the registration has been reviewed and approved, the new member receives an email informing them of the approval. Email authentication is required for membership activation and reminder emails are sent if members have not responded to the authentication request by 1 week after approval.

Members can publish detailed research profiles on the site, upload their biography, and have the opportunity to import a list of publications directly from PubMed. A member's contributions to the site are also listed on his or her profile page. Members can choose to allow other members to contact them via email functionality provided on the member profile page. Finally, members can subscribe to receive email alerts on new content by type (News, Webinars, Jobs, etc.).

COMMUNITY AND RESEARCH RESOURCES

We provide a variety of structured resources for members: Meetings and Events, Jobs, Funding Opportunities and Bulletin Board. Meetings and Events allows researchers to quickly find upcoming meetings of interest. These are listed automatically in reverse chronological order. Meetings can be linked to PRF Blogs and News stories on the event, allowing researchers to "catch-up" if they missed the actual event. Jobs provide networking between hiring institutions and applicants; Funding Opportunities highlights grants in the field and Bulletin Board is for posting *ad hoc* announcements. Together, these community resources provide content tailored to the professional needs of researchers in the pain field. All community resource items have social media tools and can be individually commented on, bookmarked, shared or recommended.

We also provide a variety of tools for creating and publishing research resources. Some are simply pages of information or collections of links to other useful resources. We also developed a structured database for genes associated with a disease or biological condition such as pain. Fields include data to fully describe the gene and details of data on variants associated with pain, with literature references. For studies using research models, the type of model is described, thus presenting a detailed overview of the research done to associate the gene with the disease. In the future, we would also like to create other resources, such as a drugs database that would serve as a central repository for information on new drugs in development and associated clinical trials.

SOCIAL MEDIA TOOLS

Social media tools are important for online collaboration. We developed or customized a large number of social media tools and incorporated them in our framework. Members can comment on or invite others to comment, share, bookmark, and recommend most content throughout the site. All content on PRF can be emailed and shared on all the popular social network tools (Facebook, Twitter, etc.) by using standard "email" and "share" modules present on every page. RSS news and Twitter feeds are available as well as an email newsletter.

To accommodate the needs of our scientific community, we made a large number of enhancements to the comment feature in Drupal. Scientific commentaries often have attachments or figures, so we developed capabilities for attaching images or documents. The comments can be formatted with a WYSIWYG editor and can be associated with more than one content item if applicable.

WEB SITE USE AND TRACKING

Websites are tracked using Google Analytics [5], which provides extensive data on how users interact with the site. We analyze data on number of visits, unique visitors, total pageviews, and views of individual pages. We also look at selected demographic data (country and city of origin), system information, and source of traffic to the site.

SEARCH AND SEMANTIC WEB

Search is implemented using the open-source enterprise Apache Lucene Solr [6] search platform. We also implemented section-specific searches. Search results can be sorted by date, relevance, number of comments or the date of the last comment. The number of search results per page can be configured by the user. Search results can be filtered using facets. The content type, date, categories etc. are presented as facets.

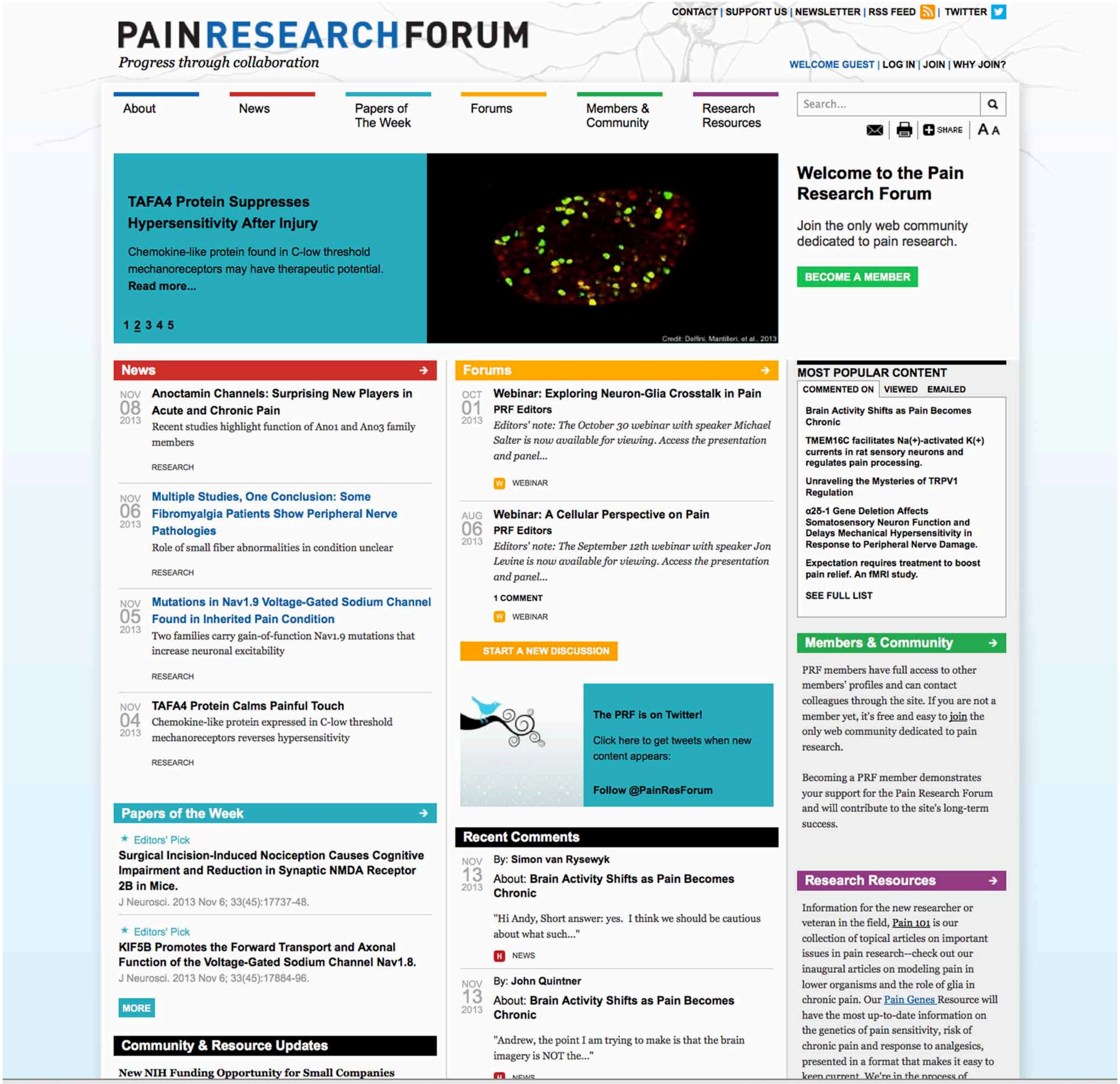
Semantic Web technologies enable publication of structured documents that can be processed by machines, thus allowing interoperability with the Web of Data (Berners-Lee et al., 2001). We use the Drupal Resource Description Framework Modules (RDF) modules (Corlosquet et al., 2009) to publish RDF of News and Forums. The RDF is indexed and stored in a SPARQL endpoint using the PHP ARC2 libraries[7]. The Dublin Core (Weibel et al., 1998) and Semantically-Interlinked Online Communities (SIOC) (Breslin et al., 2006) ontologies are used to express the RDF. SPARQL queries enable us to perform flexible queries and integrate with other knowledge repositories. Thus, incorporation of Semantic Web technologies in the SCF platform will allow us to network additional online communities built with SCF and identify relevant information across multiple sites.

RESULTS

PAIN RESEARCH FORUM

The SCF platform, originally used for publishing scholarly articles for StemBook (http://stembook.org), was reengineered to create an online community of chronic pain researchers. The goal was to accelerate pain research by enabling free discussion and faster sharing of information between academia, industry, and the clinic, to foster new collaborations and to raise interest in pain research among the wider community of neuroscientists and clinicians. The PRF [8] was launched in June 2011; a screenshot of the home page is shown in **Figure 2**. As of December 2013, PRF has attracted ∼1400 registered members and has published more than 150 News stories and 25 Discussion forums, facilitated five Webinars and published four Journal Clubs features. There are more than 200 member-authored comments on News, Papers, and other content. Papers of the Week are published every Friday and 2–6 papers are highlighted each week as Editors'

[5] http://www.google.com/analytics/
[6] http://lucene.apache.org/solr/
[7] https://github.com/semsol/arc2
[8] http://www.painresearchforum.org/



CONTACT | SUPPORT US | NEWSLETTER | RSS FEED | TWITTER

PAINRESEARCHFORUM
Progress through collaboration

WELCOME GUEST | LOG IN | JOIN | WHY JOIN?

TAFA4 Protein Suppresses Hypersensitivity After Injury

Chemokine-like protein found in C-low threshold mechanoreceptors may have therapeutic potential. **Read more...**

1 2 3 4 5

Credit: Delfini, Mantilleri, et al., 2013

Welcome to the Pain Research Forum

Join the only web community dedicated to pain research.

BECOME A MEMBER

News

NOV 08 2013 **Anoctamin Channels: Surprising New Players in Acute and Chronic Pain**
Recent studies highlight function of Ano1 and Ano3 family members
RESEARCH

NOV 06 2013 **Multiple Studies, One Conclusion: Some Fibromyalgia Patients Show Peripheral Nerve Pathologies**
Role of small fiber abnormalities in condition unclear
RESEARCH

NOV 05 2013 **Mutations in Nav1.9 Voltage-Gated Sodium Channel Found in Inherited Pain Condition**
Two families carry gain-of-function Nav1.9 mutations that increase neuronal excitability
RESEARCH

NOV 04 2013 **TAFA4 Protein Calms Painful Touch**
Chemokine-like protein expressed in C-low threshold mechanoreceptors reverses hypersensitivity
RESEARCH

Papers of the Week

★ Editors' Pick
Surgical Incision-Induced Nociception Causes Cognitive Impairment and Reduction in Synaptic NMDA Receptor 2B in Mice.
J Neurosci. 2013 Nov 6; 33(45):17737-48.

★ Editors' Pick
KIF5B Promotes the Forward Transport and Axonal Function of the Voltage-Gated Sodium Channel Nav1.8.
J Neurosci. 2013 Nov 6; 33(45):17884-96.

MORE

Community & Resource Updates

New NIH Funding Opportunity for Small Companies

Forums

OCT 01 2013 **Webinar: Exploring Neuron-Glia Crosstalk in Pain**
PRF Editors
Editors' note: The October 30 webinar with speaker Michael Salter is now available for viewing. Access the presentation and panel...
W WEBINAR

AUG 06 2013 **Webinar: A Cellular Perspective on Pain**
PRF Editors
Editors' note: The September 12th webinar with speaker Jon Levine is now available for viewing. Access the presentation and panel...
1 COMMENT
W WEBINAR

START A NEW DISCUSSION

The PRF is on Twitter!
Click here to get tweets when new content appears:
Follow @PainResForum

Recent Comments

NOV 13 2013 By: **Simon van Rysewyk**
About: **Brain Activity Shifts as Pain Becomes Chronic**
"Hi Andy, Short answer: yes. I think we should be cautious about what such..."
N NEWS

NOV 13 2013 By: **John Quintner**
About: **Brain Activity Shifts as Pain Becomes Chronic**
"Andrew, the point I am trying to make is that the brain imagery is NOT the..."
N NEWS

MOST POPULAR CONTENT

COMMENTED ON | VIEWED | EMAILED

Brain Activity Shifts as Pain Becomes Chronic

TMEM16C facilitates Na(+)-activated K(+) currents in rat sensory neurons and regulates pain processing.

Unraveling the Mysteries of TRPV1 Regulation

α2δ-1 Gene Deletion Affects Somatosensory Neuron Function and Delays Mechanical Hypersensitivity in Response to Peripheral Nerve Damage.

Expectation requires treatment to boost pain relief. An fMRI study.

SEE FULL LIST

Members & Community

PRF members have full access to other members' profiles and can contact colleagues through the site. If you are not a member yet, it's free and easy to join the only web community dedicated to pain research.

Becoming a PRF member demonstrates your support for the Pain Research Forum and will contribute to the site's long-term success.

Research Resources

Information for the new researcher or veteran in the field, Pain 101 is our collection of topical articles on important issues in pain research--check out our inaugural articles on modeling pain in lower organisms and the role of glia in chronic pain. Our Pain Genes Resource will have the most up-to-date information on the genetics of pain sensitivity, risk of chronic pain and response to analgesics, presented in a format that makes it easy to keep current. We're in the process of

FIGURE 2 | Pain Research Forum. Screenshot of home page for Pain Research Forum (http://www.painresearchforum.org) for anonymous users (not logged-in).

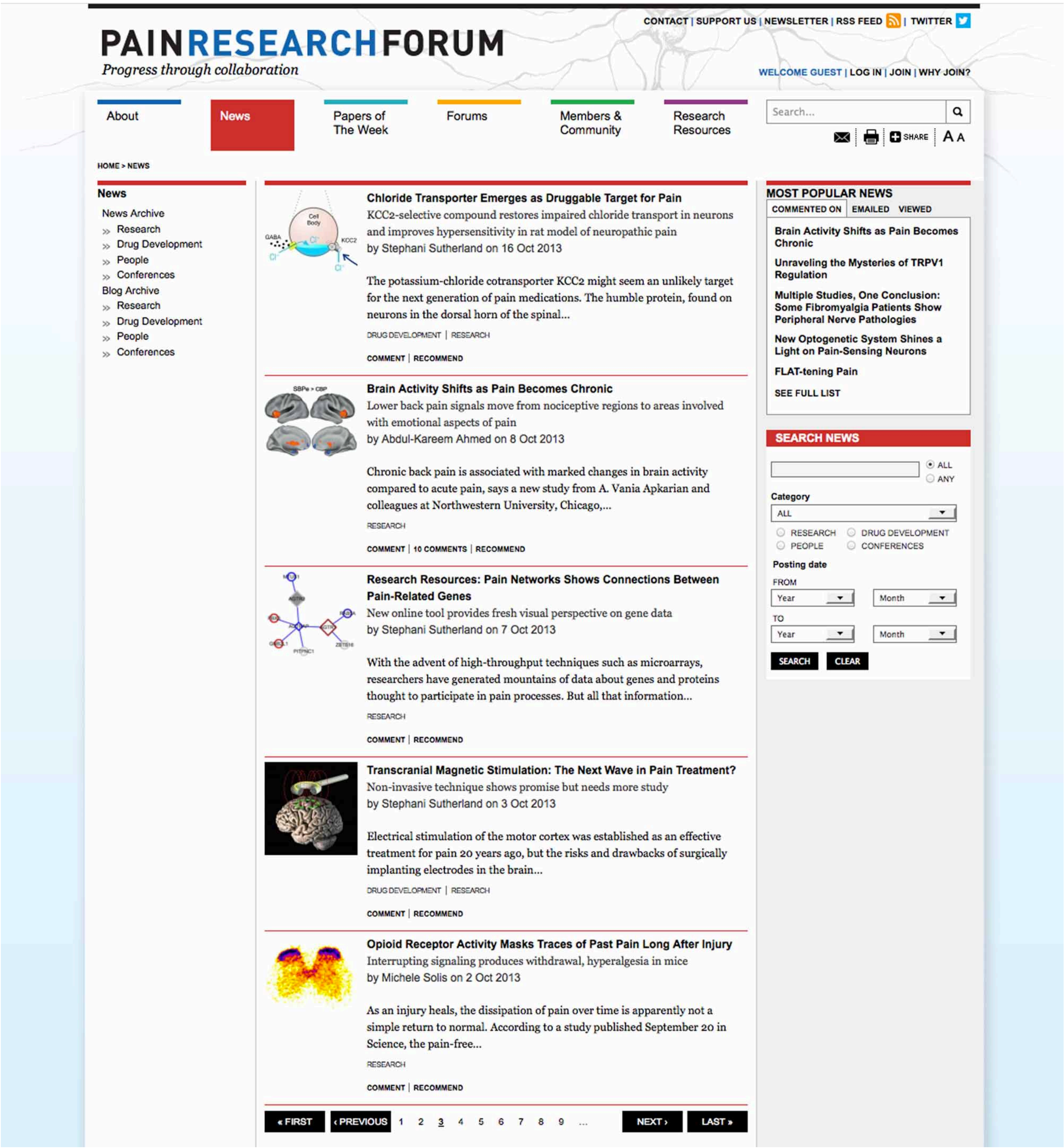
Picks. Curated and frequently updated lists of Meetings and Events, Jobs, Funding Opportunities, and Bulletin Board items are posted. The site editors actively curate and maintain several research resources. All content is licensed for reuse under Creative Commons license BY-ND-NC[9] .

NEWS

PRF publishes 1–2 news stories each week; a screen shot of the News section is shown in **Figure 3**. News stories are categorized as "Research," "Drug Development," "People" or "Conferences." PRF's news coverage helps researchers stay abreast of the latest findings in the field. For example, PRF was one of the first media outlets to publish a news story about a study of how "high-dose opioid reverses synaptic potentiation in the spinal cord in rats" (Talkington, 2012). The research paper was published in Science on January 13, 2012 and the PRF news story came out 3 days later. Four prominent pain researchers presented their opinions on the work in the form of comments to the story.

Often several related stories are published on an individual topic, and SCF is engineered to allow integration of this material.

[9] http://creativecommons.org/licenses/by-nc-nd/3.0/



CONTACT | SUPPORT US | NEWSLETTER | RSS FEED | TWITTER

PAINRESEARCHFORUM

Progress through collaboration

WELCOME GUEST | LOG IN | JOIN | WHY JOIN?

About | News | Papers of The Week | Forums | Members & Community | Research Resources

Search...

SHARE | A A

HOME > NEWS

News

News Archive
- » Research
- » Drug Development
- » People
- » Conferences

Blog Archive
- » Research
- » Drug Development
- » People
- » Conferences

Chloride Transporter Emerges as Druggable Target for Pain

KCC2-selective compound restores impaired chloride transport in neurons and improves hypersensitivity in rat model of neuropathic pain

by Stephani Sutherland on 16 Oct 2013

The potassium-chloride cotransporter KCC2 might seem an unlikely target for the next generation of pain medications. The humble protein, found on neurons in the dorsal horn of the spinal...

DRUG DEVELOPMENT | RESEARCH

COMMENT | RECOMMEND

Brain Activity Shifts as Pain Becomes Chronic

Lower back pain signals move from nociceptive regions to areas involved with emotional aspects of pain

by Abdul-Kareem Ahmed on 8 Oct 2013

Chronic back pain is associated with marked changes in brain activity compared to acute pain, says a new study from A. Vania Apkarian and colleagues at Northwestern University, Chicago,...

RESEARCH

COMMENT | 10 COMMENTS | RECOMMEND

Research Resources: Pain Networks Shows Connections Between Pain-Related Genes

New online tool provides fresh visual perspective on gene data

by Stephani Sutherland on 7 Oct 2013

With the advent of high-throughput techniques such as microarrays, researchers have generated mountains of data about genes and proteins thought to participate in pain processes. But all that information...

RESEARCH

COMMENT | RECOMMEND

Transcranial Magnetic Stimulation: The Next Wave in Pain Treatment?

Non-invasive technique shows promise but needs more study

by Stephani Sutherland on 3 Oct 2013

Electrical stimulation of the motor cortex was established as an effective treatment for pain 20 years ago, but the risks and drawbacks of surgically implanting electrodes in the brain...

DRUG DEVELOPMENT | RESEARCH

COMMENT | RECOMMEND

Opioid Receptor Activity Masks Traces of Past Pain Long After Injury

Interrupting signaling produces withdrawal, hyperalgesia in mice

by Michele Solis on 2 Oct 2013

As an injury heals, the dissipation of pain over time is apparently not a simple return to normal. According to a study published September 20 in Science, the pain-free...

RESEARCH

COMMENT | RECOMMEND

« FIRST | ‹ PREVIOUS | 1 2 3 4 5 6 7 8 9 ... | NEXT › | LAST »

MOST POPULAR NEWS

COMMENTED ON | EMAILED | VIEWED

- Brain Activity Shifts as Pain Becomes Chronic
- Unraveling the Mysteries of TRPV1 Regulation
- Multiple Studies, One Conclusion: Some Fibromyalgia Patients Show Peripheral Nerve Pathologies
- New Optogenetic System Shines a Light on Pain-Sensing Neurons
- FLAT-tening Pain

SEE FULL LIST

SEARCH NEWS

● ALL
○ ANY

Category

ALL

○ RESEARCH ○ DRUG DEVELOPMENT
○ PEOPLE ○ CONFERENCES

Posting date

FROM

Year | Month

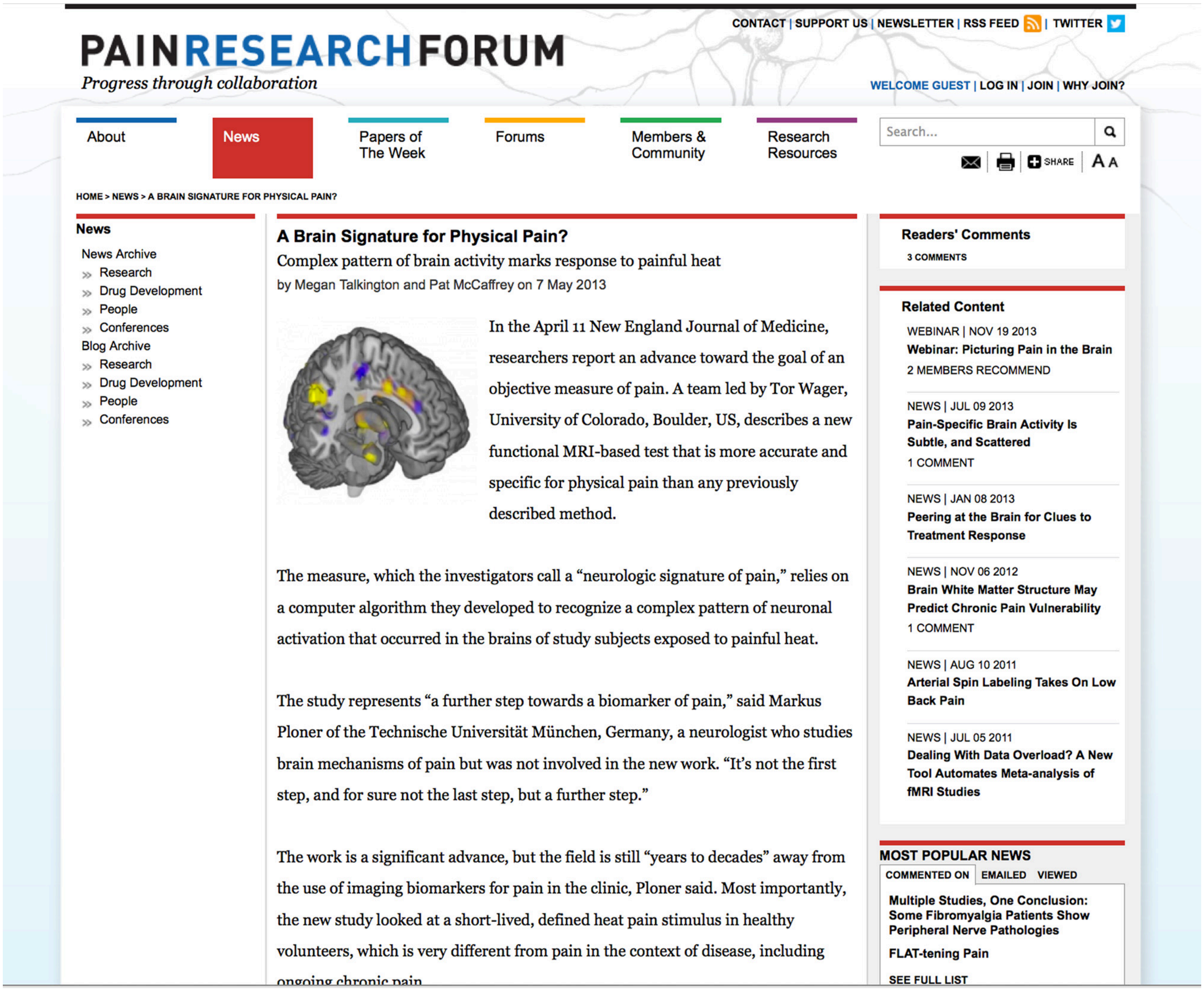
TO

Year | Month

SEARCH | CLEAR

FIGURE 3 | News Section in PRF. Screenshot of news section in Pain Research Forum (http://www.painresearchforum.org/news). Five stories are listed per page, social media tools are available for each story. News stories can be filtered using categories on left. Most popular items are highlighted on the right.

For instance, PRF recently covered three high-profile brain-imaging studies (Ahmed, 2013; Talkington, 2013; Talkington and McCaffrey, 2013). A screen shot of one of the stories is shown in **Figure 4**. The forms used to create and publish the story are shown in **Figure 5**. Related stories are listed in a block on the right. In addition, PRF conducted a webinar in December 2013, featuring one of the principal investigators on the brain imaging papers, along with several panelists including authors of the other papers mentioned in the news coverage. The webinar is also listed as related content to the news story. By hyperlinking, using the References function, cross-posting comments on both news stories and papers, and using the Related Content feature, PRF is able to provide a contextualized, intelligent overview of fast-moving developments in this corner of the larger field. Social media tools



A Brain Signature for Physical Pain?

Complex pattern of brain activity marks response to painful heat

by Megan Talkington and Pat McCaffrey on 7 May 2013

In the April 11 New England Journal of Medicine, researchers report an advance toward the goal of an objective measure of pain. A team led by Tor Wager, University of Colorado, Boulder, US, describes a new functional MRI-based test that is more accurate and specific for physical pain than any previously described method.

The measure, which the investigators call a "neurologic signature of pain," relies on a computer algorithm they developed to recognize a complex pattern of neuronal activation that occurred in the brains of study subjects exposed to painful heat.

The study represents "a further step towards a biomarker of pain," said Markus Ploner of the Technische Universität München, Germany, a neurologist who studies brain mechanisms of pain but was not involved in the new work. "It's not the first step, and for sure not the last step, but a further step."

The work is a significant advance, but the field is still "years to decades" away from the use of imaging biomarkers for pain in the clinic, Ploner said. Most importantly, the new study looked at a short-lived, defined heat pain stimulus in healthy volunteers, which is very different from pain in the context of disease, including ongoing chronic pain

Readers' Comments

3 COMMENTS

Related Content

WEBINAR | NOV 19 2013
Webinar: Picturing Pain in the Brain
2 MEMBERS RECOMMEND

NEWS | JUL 09 2013
Pain-Specific Brain Activity Is Subtle, and Scattered
1 COMMENT

NEWS | JAN 08 2013
Peering at the Brain for Clues to Treatment Response

NEWS | NOV 06 2012
Brain White Matter Structure May Predict Chronic Pain Vulnerability
1 COMMENT

NEWS | AUG 10 2011
Arterial Spin Labeling Takes On Low Back Pain

NEWS | JUL 05 2011
Dealing With Data Overload? A New Tool Automates Meta-analysis of fMRI Studies

MOST POPULAR NEWS

COMMENTED ON | EMAILED | VIEWED

Multiple Studies, One Conclusion: Some Fibromyalgia Patients Show Peripheral Nerve Pathologies

FLAT-tening Pain

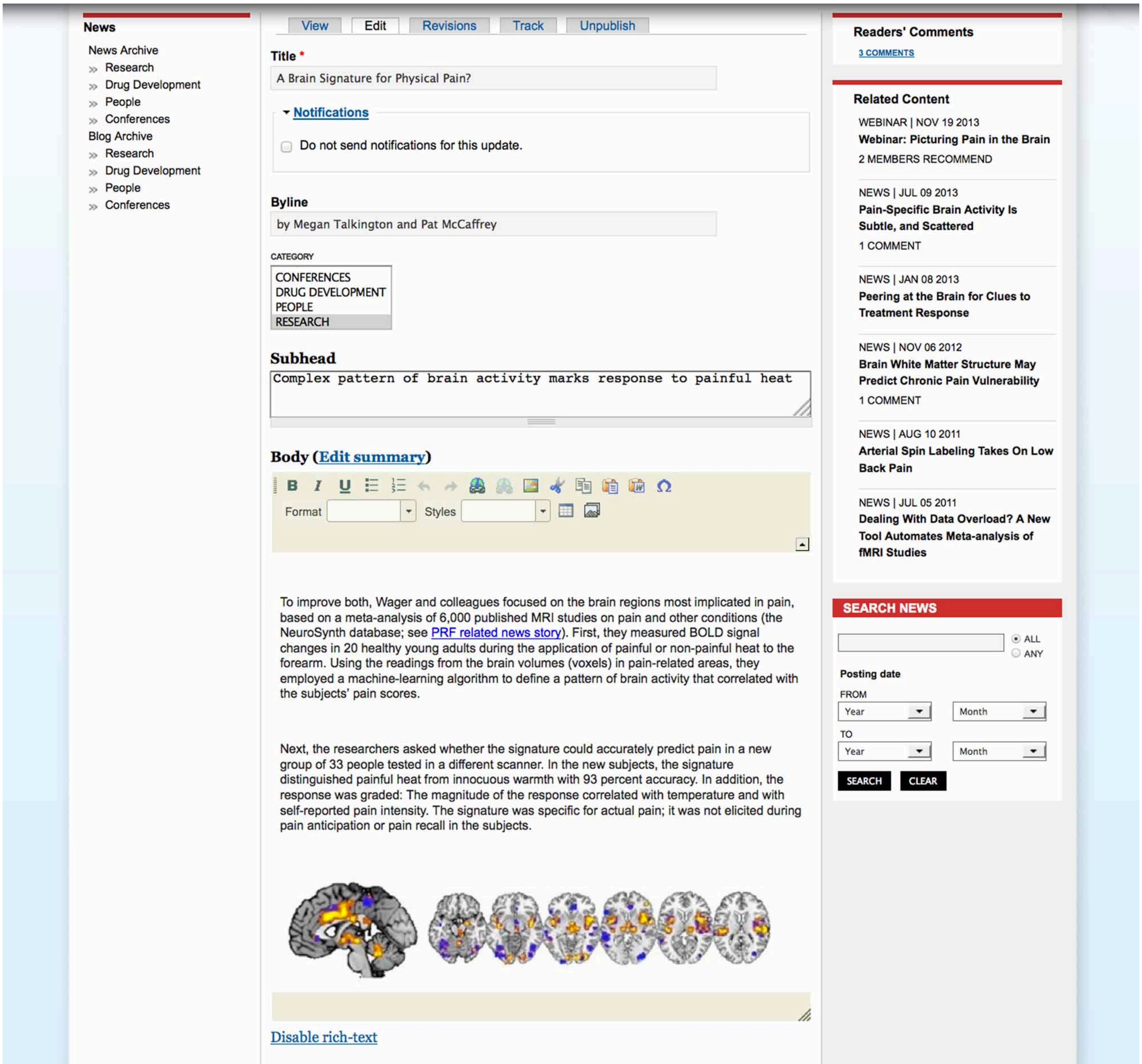
SEE FULL LIST

FIGURE 4 | News story on brain imaging. Screenshot of news story on brain imaging study (http://www.painresearchforum.org/news/27192-brain-signature-physical-pain). Study was led by Tor Wager, University of Colorado, Boulder, US and describes a new functional MRI-based test for measuring pain. Related stories are listed on a block on the right. The article has 3 comments.

such as Twitter feeds and newsletters disseminate the information quickly and effectively to members.

Brain imaging in the context of understanding and detecting pain is a popular but controversial topic, and the three stories cited above (Ahmed, 2013; Talkington, 2013; Talkington and McCaffrey, 2013) elicited several comments from PRF members. It is significant that many of the researchers commenting on PRF are junior people, including graduate students or postdoctoral fellows. Often the study authors participate in the discussion: for example, on the news story by Talkington and McCaffrey (2013), the study authors responded to two previous comments from researchers not involved with the study. Thus, the commenting feature on PRF news stories serves a function similar to the "letters to the editor" sections of journals, with key differences: it is faster, has a lower barrier to entry, and welcomes contributions from junior researchers.

Currently, the most accessed news story is one that discusses the new research on the use of antibiotics to relieve some forms of chronic lower back pain (Morton, 2013). The PRF news story covers two published research studies: one suggests that pain may be caused by a low-grade bacterial infection in the discs and the other finds that antibiotics can effectively treat the pain and prevent further tissue degeneration. Both studies have implications for patients with long-standing low back pain, and they elicited a lot of attention and controversy including in the popular press. In a Google search for the query "antibiotics for back pain," the PRF news story is the number one hit. This shows that PRF news stories can be highly ranked in Google searches for general pain terms, giving many readers access to the site. This high ranking may contribute to the unusually high number of page views for this article, which are three times more than the second most viewed news article.



Title *

A Brain Signature for Physical Pain?

▾ Notifications

□ Do not send notifications for this update.

Byline

by Megan Talkington and Pat McCaffrey

CATEGORY

CONFERENCES
DRUG DEVELOPMENT
PEOPLE
RESEARCH

Subhead

Complex pattern of brain activity marks response to painful heat

Body (Edit summary)

To improve both, Wager and colleagues focused on the brain regions most implicated in pain, based on a meta-analysis of 6,000 published MRI studies on pain and other conditions (the NeuroSynth database; see PRF related news story). First, they measured BOLD signal changes in 20 healthy young adults during the application of painful or non-painful heat to the forearm. Using the readings from the brain volumes (voxels) in pain-related areas, they employed a machine-learning algorithm to define a pattern of brain activity that correlated with the subjects' pain scores.

Next, the researchers asked whether the signature could accurately predict pain in a new group of 33 people tested in a different scanner. In the new subjects, the signature distinguished painful heat from innocuous warmth with 93 percent accuracy. In addition, the response was graded: The magnitude of the response correlated with temperature and with self-reported pain intensity. The signature was specific for actual pain; it was not elicited during pain anticipation or pain recall in the subjects.

Readers' Comments

3 COMMENTS

Related Content

WEBINAR | NOV 19 2013
Webinar: Picturing Pain in the Brain
2 MEMBERS RECOMMEND

NEWS | JUL 09 2013
Pain-Specific Brain Activity Is Subtle, and Scattered
1 COMMENT

NEWS | JAN 08 2013
Peering at the Brain for Clues to Treatment Response

NEWS | NOV 06 2012
Brain White Matter Structure May Predict Chronic Pain Vulnerability
1 COMMENT

NEWS | AUG 10 2011
Arterial Spin Labeling Takes On Low Back Pain

NEWS | JUL 05 2011
Dealing With Data Overload? A New Tool Automates Meta-analysis of fMRI Studies

FIGURE 5 | Forms used by editors for a News story. Screenshot of forms used by the editors to create and publish a news story. A form is available for each field and the body is composed using a WYSIWYG editor.

FORUMS

The PRF Forums category includes Discussions, Webinars, and Journal Clubs. PRF editors have moderated several online discussions. The most-accessed Discussion is a debate on how the human brain processes stimuli, initiated by a well known pain researcher and PRF Science Advisory Board member (Basbaum, 2011). A Discussion on the challenges associated with translating pain research discoveries into clinical developments, presented by another researcher and science advisor, attracted many follow-on comments (Mogil, 2011). In 2013, PRF conducted five Webinars, each typically attracting ∼150 registered attendees, plus an unknown number of additional viewers who watched the event in groups under one registration. Each Webinar is conducted online using a webinar-hosting service and a recording is subsequently posted to PRF with a written introduction. This archives the presentation for future viewing and enables an ongoing, online conversation beyond the duration of the actual presentation. The Journal Club is a venue for disseminating the results of discussions that occur in individual lab groups about recently published scientific articles. For example, a graduate student and postdoctoral fellow studying pediatric pain presented a journal club at their institution on Walker et al. (2012; Birnie and Caes, 2012). They then wrote for PRF a brief synopsis of the study and the discussion that took place in their meeting. They also posed questions to the author of the original paper, who responded with her own online comment. A screenshot of the Journal Club is shown in **Figure 6**.

PAPERS OF THE WEEK

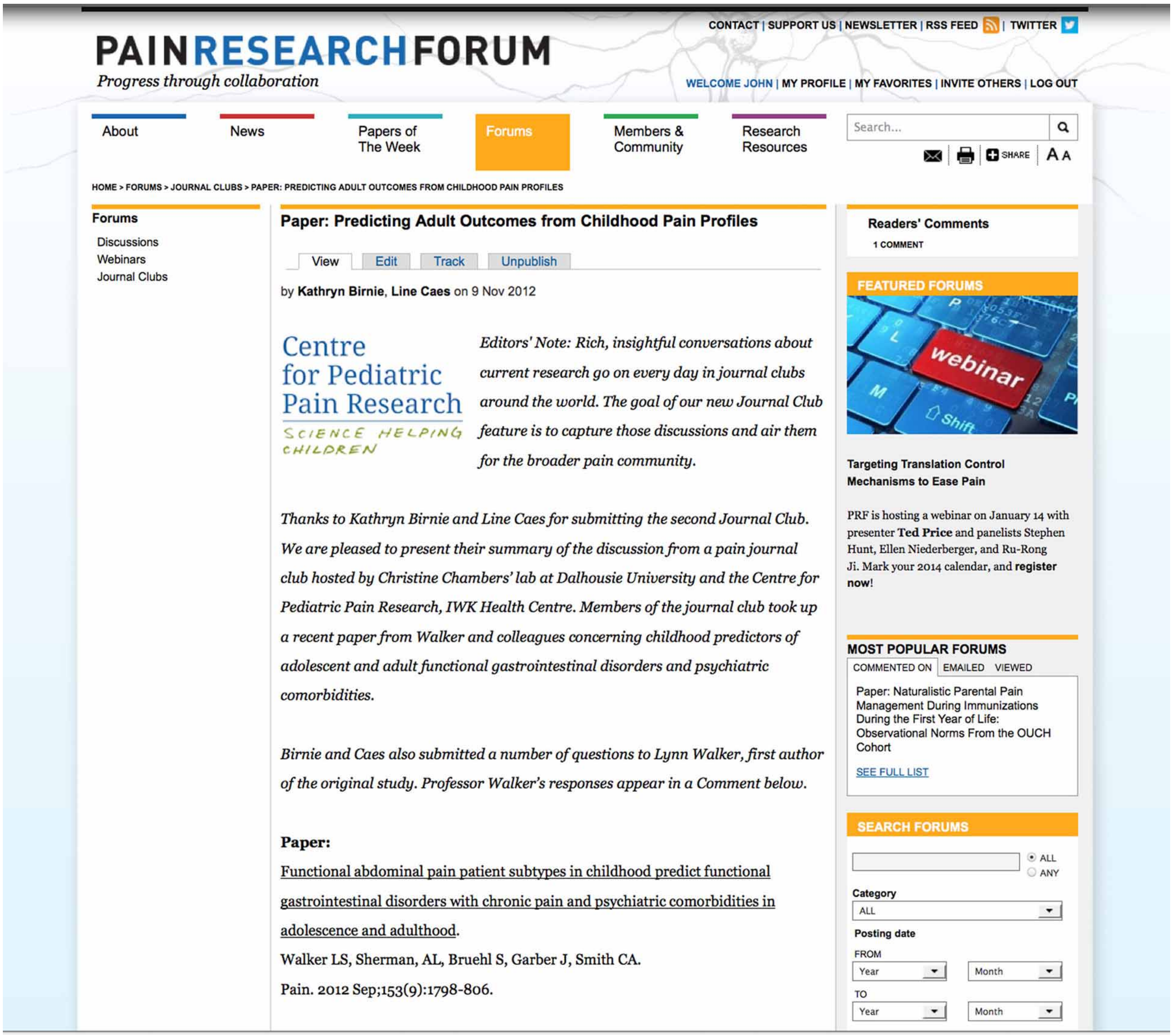
Papers of the Week are published every Friday and provide a curated digest of recent and noteworthy pain-relevant articles published in academic journals. For example, for the November 2–8, 2013 collection[10] , 61 papers were identified and listed. Two were further highlighted as Editors' Picks. Highlighted papers are often commented on by PRF-registered members, sometimes resulting in vibrant back-and-forth discussions between the authors and other PRF members[11] . Related papers are listed in a block on the right as shown in **Figure 7**; a news story on the paper is listed under the paper citation. Papers of the Week are archived as weekly lists, and the database of all papers can be searched with a detailed section-specific search. Members can also search for other papers using links provided on the paper page to Google Scholar or PubMed.

PRF MEMBERSHIP AND USAGE STATISTICS

As of December 17, 2013, PRF has just over 1400 registered members, of whom 1143 have published profiles in the member directory. Member demographics are shown in **Figure 8**. A variety of professionals sectors are represented including universities, hospitals, industry, government, and non-profit foundations, with the majority of members coming from academia (**Figure 8A**). Most members are research scientists and academicians including graduate students and postdoctoral fellows (**Figure 8B**). Two-thirds (67%) of PRF members have an advanced degree, 41% have earned a PhD and 20% have an MD or DDS.

[10] http://www.painresearchforum.org/node/33603

[11] http://www.painresearchforum.org/papers/22720-activation-5-ht2a-receptors-upregulates-function-neuronal-k-cl-cotransporter-kcc2



Paper: Predicting Adult Outcomes from Childhood Pain Profiles

by **Kathryn Birnie**, **Line Caes** on 9 Nov 2012

Editors' Note: Rich, insightful conversations about current research go on every day in journal clubs around the world. The goal of our new Journal Club feature is to capture those discussions and air them for the broader pain community.

Thanks to Kathryn Birnie and Line Caes for submitting the second Journal Club. We are pleased to present their summary of the discussion from a pain journal club hosted by Christine Chambers' lab at Dalhousie University and the Centre for Pediatric Pain Research, IWK Health Centre. Members of the journal club took up a recent paper from Walker and colleagues concerning childhood predictors of adolescent and adult functional gastrointestinal disorders and psychiatric comorbidities.

Birnie and Caes also submitted a number of questions to Lynn Walker, first author of the original study. Professor Walker's responses appear in a Comment below.

Paper:

Functional abdominal pain patient subtypes in childhood predict functional gastrointestinal disorders with chronic pain and psychiatric comorbidities in adolescence and adulthood.

Walker LS, Sherman, AL, Bruehl S, Garber J, Smith CA.

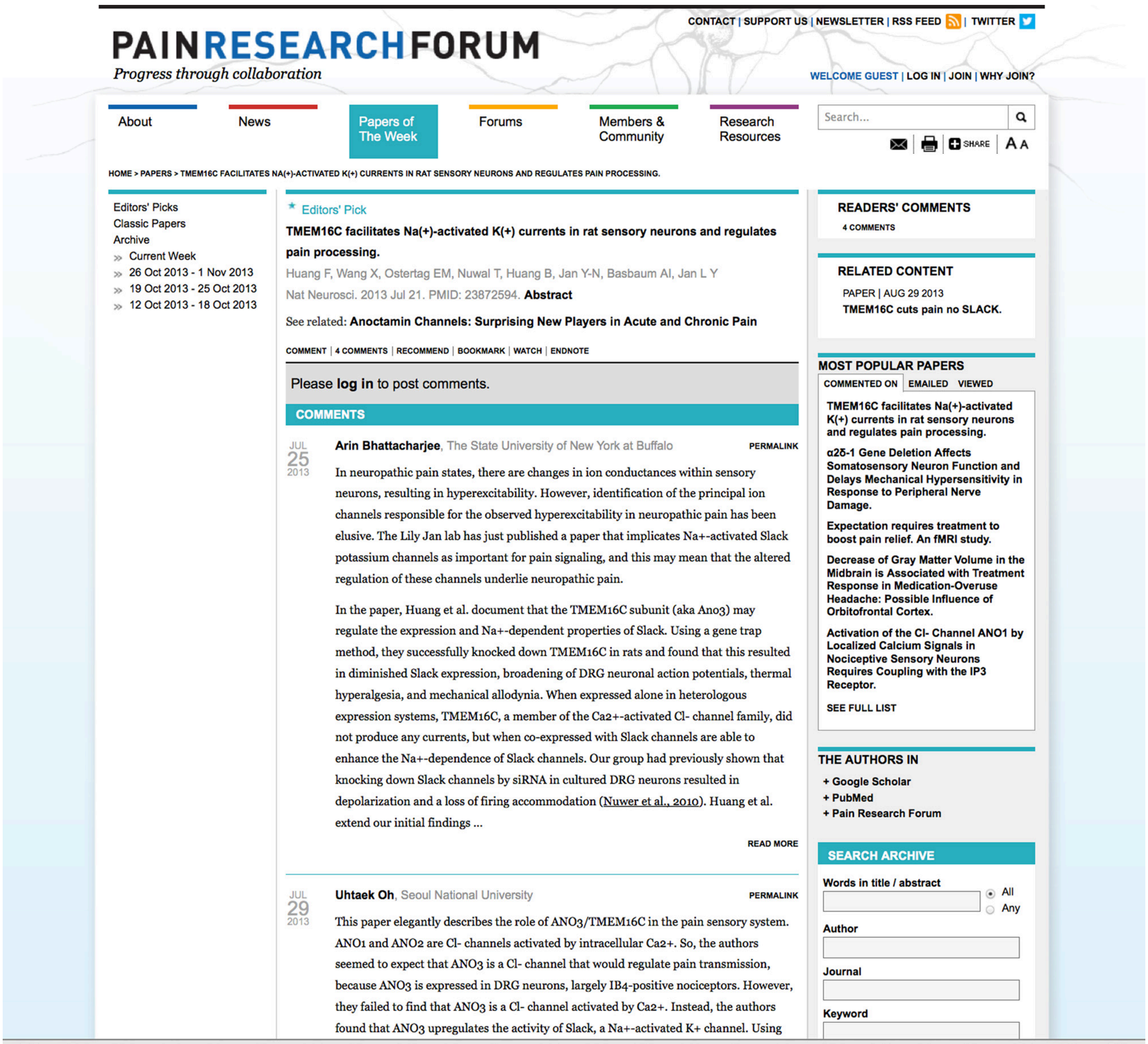
Pain. 2012 Sep;153(9):1798-806.

Targeting Translation Control Mechanisms to Ease Pain

PRF is hosting a webinar on January 14 with presenter **Ted Price** and panelists Stephen Hunt, Ellen Niederberger, and Ru-Rong Ji. Mark your 2014 calendar, and **register now!**

Paper: Naturalistic Parental Pain Management During Immunizations During the First Year of Life: Observational Norms From the OUCH Cohort

FIGURE 6 | Journal Club on recently published paper. Screenshot of journal club featuring a recently published paper in the journal PAIN (http://www.painresearchforum.org/forums/journal-club/21586-predicting-adult-outcomes-childhood-pain-profiles). Featured forums are highlighted in a block on the right.



CONTACT | SUPPORT US | NEWSLETTER | RSS FEED | TWITTER

PAINRESEARCHFORUM
Progress through collaboration

WELCOME GUEST | LOG IN | JOIN | WHY JOIN?

HOME > PAPERS > TMEM16C FACILITATES NA(+)-ACTIVATED K(+) CURRENTS IN RAT SENSORY NEURONS AND REGULATES PAIN PROCESSING.

Editors' Picks
Classic Papers
Archive
» Current Week
» 26 Oct 2013 - 1 Nov 2013
» 19 Oct 2013 - 25 Oct 2013
» 12 Oct 2013 - 18 Oct 2013

★ Editors' Pick

TMEM16C facilitates Na(+)-activated K(+) currents in rat sensory neurons and regulates pain processing.

Huang F, Wang X, Ostertag EM, Nuwal T, Huang B, Jan Y-N, Basbaum AI, Jan L Y

Nat Neurosci. 2013 Jul 21. PMID: 23872594. **Abstract**

See related: **Anoctamin Channels: Surprising New Players in Acute and Chronic Pain**

COMMENT | 4 COMMENTS | RECOMMEND | BOOKMARK | WATCH | ENDNOTE

Please **log in** to post comments.

COMMENTS

JUL 25 2013 **Arin Bhattacharjee**, The State University of New York at Buffalo PERMALINK

In neuropathic pain states, there are changes in ion conductances within sensory neurons, resulting in hyperexcitability. However, identification of the principal ion channels responsible for the observed hyperexcitability in neuropathic pain has been elusive. The Lily Jan lab has just published a paper that implicates Na+-activated Slack potassium channels as important for pain signaling, and this may mean that the altered regulation of these channels underlie neuropathic pain.

In the paper, Huang et al. document that the TMEM16C subunit (aka Ano3) may regulate the expression and Na+-dependent properties of Slack. Using a gene trap method, they successfully knocked down TMEM16C in rats and found that this resulted in diminished Slack expression, broadening of DRG neuronal action potentials, thermal hyperalgesia, and mechanical allodynia. When expressed alone in heterologous expression systems, TMEM16C, a member of the Ca2+-activated Cl- channel family, did not produce any currents, but when co-expressed with Slack channels are able to enhance the Na+-dependence of Slack channels. Our group had previously shown that knocking down Slack channels by siRNA in cultured DRG neurons resulted in depolarization and a loss of firing accommodation (Nuwer et al., 2010). Huang et al. extend our initial findings ...

READ MORE

JUL 29 2013 **Uhtaek Oh**, Seoul National University PERMALINK

This paper elegantly describes the role of ANO3/TMEM16C in the pain sensory system. ANO1 and ANO2 are Cl- channels activated by intracellular Ca2+. So, the authors seemed to expect that ANO3 is a Cl- channel that would regulate pain transmission, because ANO3 is expressed in DRG neurons, largely IB4-positive nociceptors. However, they failed to find that ANO3 is a Cl- channel activated by Ca2+. Instead, the authors found that ANO3 upregulates the activity of Slack, a Na+-activated K+ channel. Using

READERS' COMMENTS
4 COMMENTS

RELATED CONTENT
PAPER | AUG 29 2013
TMEM16C cuts pain no SLACK.

MOST POPULAR PAPERS
COMMENTED ON EMAILED VIEWED

TMEM16C facilitates Na(+)-activated K(+) currents in rat sensory neurons and regulates pain processing.

α2δ-1 Gene Deletion Affects Somatosensory Neuron Function and Delays Mechanical Hypersensitivity in Response to Peripheral Nerve Damage.

Expectation requires treatment to boost pain relief. An fMRI study.

Decrease of Gray Matter Volume in the Midbrain is Associated with Treatment Response in Medication-Overuse Headache: Possible Influence of Orbitofrontal Cortex.

Activation of the Cl- Channel ANO1 by Localized Calcium Signals in Nociceptive Sensory Neurons Requires Coupling with the IP3 Receptor.

SEE FULL LIST

THE AUTHORS IN
+ Google Scholar
+ PubMed
+ Pain Research Forum

SEARCH ARCHIVE
Words in title / abstract ● All ○ Any
Author
Journal
Keyword

FIGURE 7 | Sample Paper from Papers of the Week. Screenshot of paper published in Papers of the Week (http://www.painresearchforum.org/papers/29791-tmem16c-facilitates-na-activated-k-currents-rat-sensory-neurons-and-regulates-pain). Related paper is listed on the right. Section specific search is also available.

Although promotional efforts—emails, conference attendance, printed literature, etc.—have been responsible for attracting many members (**Figure 8C**), more than half of members found PRF either via a Web search, word of mouth or were directly invited to join by an existing member. Social media tools played an important role in recruiting members ("Other" Category).

We found that a significant proportion (about one-third) of newly registered members failed to respond to the email validation message, which asked them to click a link and return to the site to complete the registration process. We installed a module to automatically send reminder emails to these new members, and have recovered about half of the non-responders.

Google Analytics shows that in October 2013, PRF had ∼8000 unique visitors, ∼11,000 visits and over 25,000 page views. PRF has visitors from all over the world, with the majority from the USA, UK, Canada, and Australia. The most popular browsers are Chrome, Safari, and Firefox. The Papers of the Week and News are the most popular sections; popular community pages include Jobs and Meetings & Events. Some of the news stories mentioned above are among the top pages visited on the site.

COMMUNITY AND RESEARCH RESOURCES

PRF lists community and research resources of interest to the community, and the software automatically highlights new listings on the home page and section landing pages. Currently users

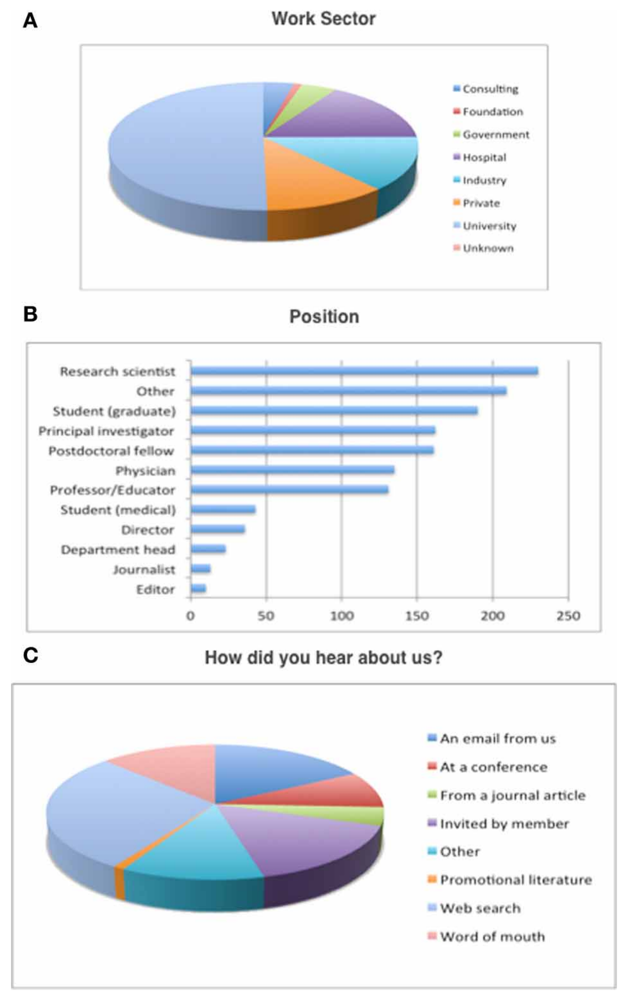


FIGURE 8 | Membership Statistics. (A) Shows the distribution of work sectors, **(B)** shows the different position of members, and **(C)** depicts the how different members heard about PRF.

can access information on more than 30 upcoming meetings and read coverage of past meetings. Job postings[12], funding opportunities[13], and bulletin board[14] items are posted. In addition, several research resources are also provided including a curated pain gene resource consisting of about 25 genes that are associated with pain, according to peer reviewed, published studies, and a collection of "Pain 101" articles covering basic questions in pain research. A collection of useful links is planned.

SEARCH AND SEMANTIC WEB

We have implemented a general as well as section-specific search for PRF using the Apace Solr module as described in the Methods section. The search results can be further filtered by content type, date, news topic or whether an item is recommended or has comments (**Figure 9**). The search term is highlighted in the results and users have the option to sort the results by type, date, number of comments or last comment date.

We have created Linked data and RDF for News, Forums, and Papers using the Drupal RDF modules along with the SIOC and DC ontologies. A sample RDF[15] illustrates the use of these ontologies to describe the News article. The RDF is indexed using the ARC2 PHP libraries and is available at http://www.painresearchforum.org/sparql. The SPARQL endpoint allows us to perform flexible queries such as "all News articles published with greater than 2 comments" (see **Table 1**). In the future, we could perform federated queries across endpoints from multiple communities.

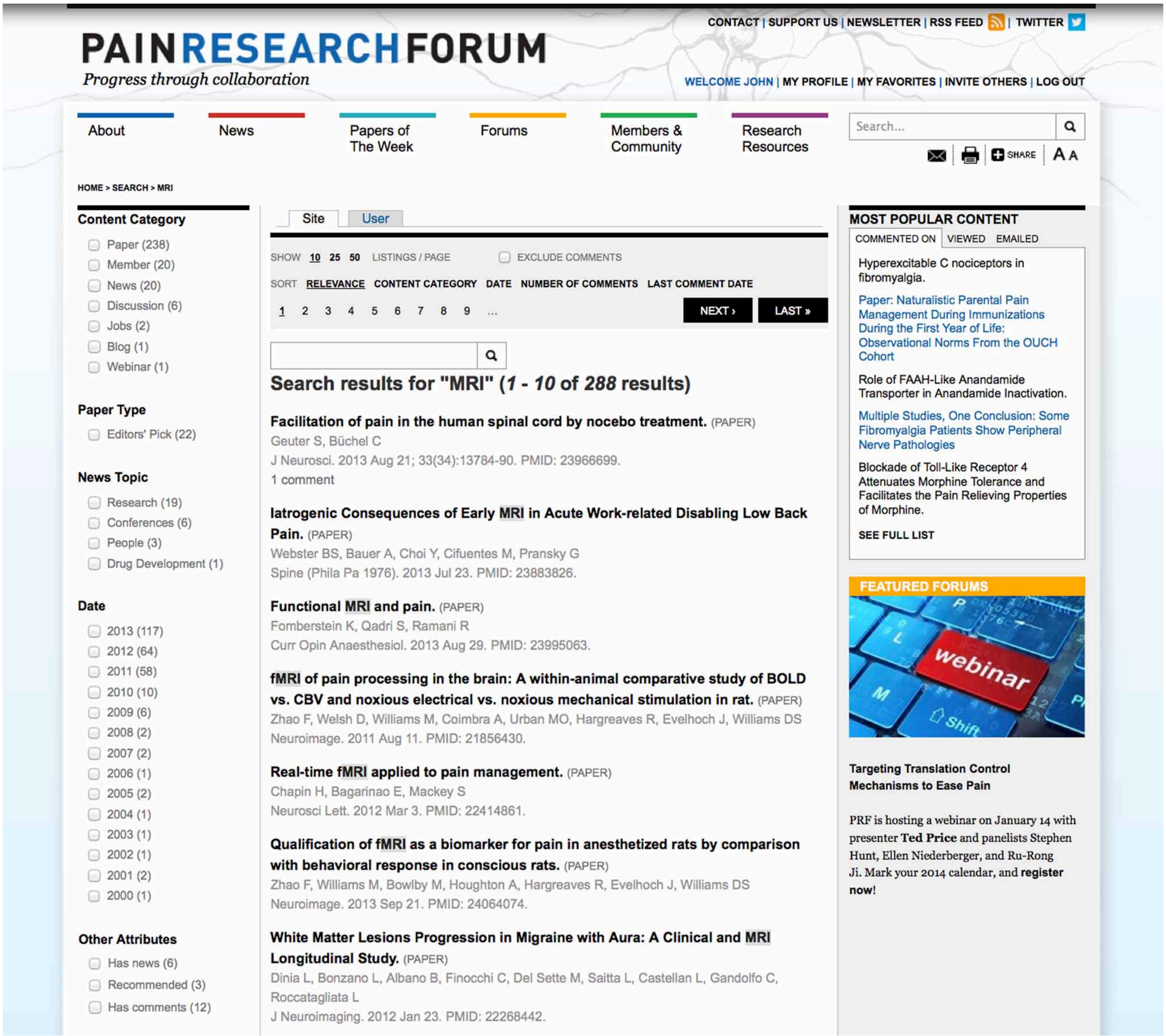
DISCUSSION

Our reusable platform, SCF, was successfully deployed to create a vibrant online community of chronic pain researchers: PRF. In a little over 2 years, PRF has attracted a large community of registered users and contributors including members from academia, industry, government, and non-profit organizations. Scientists engaged in laboratory research, clinicians, students, and fellows are all represented. PRF is a network of investigators from various sectors and disciplines and a venue for discussing, critiquing, and advancing pain research. The response to PRF in terms of member registrations and site use indicates a pent-up demand for such online communities that provide researchers in disease-circumscribed fields of biomedical research with news, forums, and resources that are most relevant to them in one place.

In our experience, editorial involvement is crucial to keep the site active and vibrant with user interactions. Reporting news, moderating discussions, soliciting comments and producing webinars and journal club pieces is labor intensive but necessary to maintain high quality interactions within the user community. On PRF, this is accomplished by a staff of professional editors and writers with backgrounds in research and neurobiology, whose primary responsibility is to create and moderate content. Assembling an active and engaged Scientific Advisory Board made up of leading researchers and clinicians from a variety of disciplines is also important to ensure the highest quality content, provide community outreach and promote community involvement.

PRF is publicized in the pain research community by several avenues. The launch was announced with a press release and with a direct email to several thousand pain researchers, identified through meeting rosters, publications, and a existing pain research listserve. In addition, fliers were distributed at pain meetings and neuroscience meetings, including 7000 postcards placed in meeting bags at the most recent World Congress on Pain in Milan (2012), the largest gathering of pain researchers in the world. Several pain professional groups promote the site to their members on their websites or in member newsletters. PRF editors have given talks and posters at pain conferences. The PRF science advisors promote the site to their colleagues using slides and other materials provided by PRF. The site is also marketed through word of mouth, a monthly email newsletter, and RSS and Twitter (@PainResForum) feeds.

[12] http://www.painresearchforum.org/members/jobs
[13] http://www.painresearchforum.org/members/funding-opportunities
[14] http://www.painresearchforum.org/members/bulletin-board
[15] http://www.painresearchforum.org/node/34289.rdf



CONTACT | SUPPORT US | NEWSLETTER | RSS FEED | TWITTER

PAINRESEARCHFORUM

Progress through collaboration

WELCOME JOHN | MY PROFILE | MY FAVORITES | INVITE OTHERS | LOG OUT

About | News | Papers of The Week | Forums | Members & Community | Research Resources

Search...

SHARE A A

HOME > SEARCH > MRI

Content Category

- □ Paper (238)
- □ Member (20)
- □ News (20)
- □ Discussion (6)
- □ Jobs (2)
- □ Blog (1)
- □ Webinar (1)

Paper Type

- □ Editors' Pick (22)

News Topic

- □ Research (19)
- □ Conferences (6)
- □ People (3)
- □ Drug Development (1)

Date

- □ 2013 (117)
- □ 2012 (64)
- □ 2011 (58)
- □ 2010 (10)
- □ 2009 (6)
- □ 2008 (2)
- □ 2007 (2)
- □ 2006 (1)
- □ 2005 (2)
- □ 2004 (1)
- □ 2003 (1)
- □ 2002 (1)
- □ 2001 (2)
- □ 2000 (1)

Other Attributes

- □ Has news (6)
- □ Recommended (3)
- □ Has comments (12)

Site | User

SHOW 10 25 50 LISTINGS / PAGE □ EXCLUDE COMMENTS

SORT RELEVANCE CONTENT CATEGORY DATE NUMBER OF COMMENTS LAST COMMENT DATE

1 2 3 4 5 6 7 8 9 ... NEXT › LAST »

Search results for "MRI" (*1 - 10* of *288* results)

Facilitation of pain in the human spinal cord by nocebo treatment. (PAPER)
Geuter S, Büchel C
J Neurosci. 2013 Aug 21; 33(34):13784-90. PMID: 23966699.
1 comment

Iatrogenic Consequences of Early MRI in Acute Work-related Disabling Low Back Pain. (PAPER)
Webster BS, Bauer A, Choi Y, Cifuentes M, Pransky G
Spine (Phila Pa 1976). 2013 Jul 23. PMID: 23883826.

Functional MRI and pain. (PAPER)
Fomberstein K, Qadri S, Ramani R
Curr Opin Anaesthesiol. 2013 Aug 29. PMID: 23995063.

fMRI of pain processing in the brain: A within-animal comparative study of BOLD vs. CBV and noxious electrical vs. noxious mechanical stimulation in rat. (PAPER)
Zhao F, Welsh D, Williams M, Coimbra A, Urban MO, Hargreaves R, Evelhoch J, Williams DS
Neuroimage. 2011 Aug 11. PMID: 21856430.

Real-time fMRI applied to pain management. (PAPER)
Chapin H, Bagarinao E, Mackey S
Neurosci Lett. 2012 Mar 3. PMID: 22414861.

Qualification of fMRI as a biomarker for pain in anesthetized rats by comparison with behavioral response in conscious rats. (PAPER)
Zhao F, Williams M, Bowlby M, Houghton A, Hargreaves R, Evelhoch J, Williams DS
Neuroimage. 2013 Sep 21. PMID: 24064074.

White Matter Lesions Progression in Migraine with Aura: A Clinical and MRI Longitudinal Study. (PAPER)
Dinia L, Bonzano L, Albano B, Finocchi C, Del Sette M, Saitta L, Castellan L, Gandolfo C, Roccatagliata L
J Neuroimaging. 2012 Jan 23. PMID: 22268442.

MOST POPULAR CONTENT

COMMENTED ON | VIEWED | EMAILED

Hyperexcitable C nociceptors in fibromyalgia.

Paper: Naturalistic Parental Pain Management During Immunizations During the First Year of Life: Observational Norms From the OUCH Cohort

Role of FAAH-Like Anandamide Transporter in Anandamide Inactivation.

Multiple Studies, One Conclusion: Some Fibromyalgia Patients Show Peripheral Nerve Pathologies

Blockade of Toll-Like Receptor 4 Attenuates Morphine Tolerance and Facilitates the Pain Relieving Properties of Morphine.

SEE FULL LIST

FEATURED FORUMS

Targeting Translation Control Mechanisms to Ease Pain

PRF is hosting a webinar on January 14 with presenter **Ted Price** and panelists Stephen Hunt, Ellen Niederberger, and Ru-Rong Ji. Mark your 2014 calendar, and **register now!**

FIGURE 9 | Search Results. This figure displays the search results for the term MRI. The search results can be filtered using various facets on the left and sorted by content type, date, number of comments or date of last comment.

Table 1 | Example SPARQL Query.

```
PREFIX schema: <http://schema.org/>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
SELECT ?post ?title ?replies
WHERE {
?post a schema:NewsArticle;
schema:name ?title;
sioc:num_replies ?replies.
FILTER ($replies > 2)
}
ORDER BY DESC(?replies)
```

Query to find news articles with > 2 comments.

One barrier to progress in research is the reticence of researchers to divulge unpublished or other preliminary work, or to publicly criticize the work of others. Web communities like PRF provide a new model of open communication that will help change this culture and promote faster and freer information exchange. A barrier to achieving more researcher involvement in and contributions to communities like PRF is the lack of incentives for scientists to contribute comments or other materials that do not add to their official publication record. In the future, sites like PRF should aim to provide incentives for contribution, for example by indexing content on PubMed or by arranging to provide continuing medical education (CME) credits for physicians who contribute.

All content on PRF is provided free of charge to the research community, and funding of shared resources like PRF is an ongoing challenge. To attract donors and other sponsors of the site we must continually demonstrate both scientific credibility in all of the content presented, and constant, lively and intellectually valuable interactions. Consistent outreach to and education of potential funders in the philanthropic area, professional societies, relevant pharmaceutical and biotech companies and the academic sector is also required. At the same time, editorial independence from sponsors must be strictly maintained. PRF does not accept paid advertisements and does not intend to do so.

In terms of technology, the SCF platform consists of a comprehensive set of building blocks for an online community. We have effectively used the SCF platform to create other communities: the Multiple Sclerosis Discovery Forum (msdiscovery.org) and StemBook (stembook.org). The platform incorporates elements of Semantic Web technologies, which have the potential to accelerate the pace of inter-disciplinary research by defining a common language and improving interoperability between various resources on the Web (Hendler, 2003). In the future, we plan to leverage these Semantic Web technologies to enable us to do cross-site queries and find relevant information on other sites. Interoperability between these multiple communities involved in neurobiology and neurology research will, we hope, identify common biological mechanisms behind complex neurological diseases and accelerate translation of science to new treatments.

ACKNOWLEDGMENTS

We gratefully acknowledge essential funding from the Curvey Family Foundation, the Mayday Fund, the Guthart Family Foundation, Purdue Pharma L.P. and Genentech. We thank Alzforum and PDOnline editors for their feedback and ideas, Nick Maloney, Paul Northup, Jake Strawn, and Chris Akeley for their contributions to software development, and Stoltze Design for graphic design.

REFERENCES

Ahmed, A. K. (2013). *Brain Activity Shifts as Pain Becomes Chronic* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/news/32409-brain-activity-shifts-pain-becomes-chronic (Accessed 2013).

Basbaum, A. (2011). *Specificity Versus Patterning Theory: Continuing the Debate* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/forums/discussion/7347-specificity-versus-patterning-theory-continuing-debate (Accessed 2013).

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Sci. Am.* 284, 28–37. doi: 10.1038/scientificamerican0501-34

Birnie, K., and Caes, L. (2012). *Paper: Predicting Adult Outcomes from Childhood Pain Profiles* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/forums/journal-club/21586-predicting-adult-outcomes-childhood-pain-profiles (Accessed 2013).

Bos, N., Zimmerman, A., Olson, J., Yew, J., Yerkie, J., Dahl, E., et al. (2007). From shared databases to communities of practice: a taxonomy of collaboratories. *J. Comput. Mediat. Comm.* 12, 652–672. doi: 10.1111/j.1083-6101.2007.00343.x

Breslin, J. G., Decker, S., Harth, A., and Bojars, U. (2006). SIOC: an approach to connect web-based communities. *Int. J. Web Based Commun.* 2, 133–142. doi: 10.1504/IJWBC.2006.010305

Clark, T., and Kinoshita, J. (2007). Alzforum and SWAN: the present and future of scientific web communities. *Brief. Bioinform.* 8, 163–171. doi: 10.1093/bib/bbm012

Corlosquet, S., Delbru, R., Clark, T. W., Polleres, A., and Decker, S. (2009). "Produce and consume linked data with drupal," in *8th International Semantic Web Conference (ISWCC)* (Washington, DC).

Das, S., Girard, L., Green, T., Weitzman, L., Lewis-Bowen, A., and Clark, T. (2009). Building biomedical web communities using a semantically aware content management system. *Brief. Bioinform.* 10, 129–138. doi: 10.1093/bib/bbn052

Finholt, T. A., and Olson, G. M. (1997). From laboratories to collaboratories: a new organizational form for scientific collaboration. *Psychol. Sci.* 8, 28–36. doi: 10.1111/j.1467-9280.1997.tb00540.x

Hendler, J. (2003). Communication. Science and the semantic web. *Science* 299, 520–521. doi: 10.1126/science.1078874

Hunter, L., and Cohen, K. B. (2006). Biomedical language processing: what's beyond PubMed? *Mol. Cell* 21, 589–594. doi: 10.1016/j.molcel.2006.02.012

Kinoshita, J., and Clark, T. (2007). Alzforum. *Methods Mol. Biol.* 401, 365–381. doi: 10.1007/978-1-59745-520-6_19

Kouzes, R. T., Myers, J. D., and Wulf, W. A. (1996). Collaboratories: doing science on the internet. *Computer* 29, 40–46. doi: 10.1109/2.532044

Kumar, K. (2012). *Drupal 7 Theming Cookbook.* Birmingham: Packt Publishing Ltd.

Mogil, J. (2011). *What Is the Reason for Lack of Translation in the Pain Field?* [Online]. Pain Research Forum. Available online at: http://www. painresearchforum.org/forums/discussion/4561-what-reason-lack-translation-pain-field (Accessed 2013).

Morton, C. C. (2013). *Antibiotics May Relieve Some Chronic Low Back Pain, Study Suggests* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/news/28081-antibiotics-may-relieve-some-chronic-low-back-pain-study-suggests (Accessed 2013).

National Research Council, USA. (1993). *National Collaboratories: Applying Information Technology for Scientific Research.* Washington, DC: The National Academy Press.

National Research Council, USA. (2011). *Relieving Pain in America: A Blueprint for Transforming Prevention, Care, Education, and Research*. Washington, DC: The National Academies Press.

Sayers, E. (2008). *E-utilities Quick Start.* Bethesda, MD: National Center for Biotechnology Information.

Talkington, M. (2012). *Erasing the Spinal Memory of Pain?* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/news/12689-erasing-spinal-memory-pain (Accessed 2013).

Talkington, M. (2013). *Pain-Specific Brain Activity is Subtle, and Scattered* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/news/29430-pain-specific-brain-activity-subtle-and-scattered (Accessed 2013).

Talkington, M., and McCaffrey, P. G. (2013). *A Brain Signature for Physical Pain?* [Online]. Pain Research Forum. Available online at: http://www.painresearchforum.org/news/27192-brain-signature-physical-pain (Accessed 2013).

Walker, L. S., Sherman, A. L., Bruehl, S., Garber, J., and Smith, C. A. (2012). Functional abdominal pain patient subtypes in childhood predict functional gastrointestinal disorders with chronic pain and psychiatric comorbidities in adolescence and adulthood. *Pain* 153, 1798–1806. doi: 10.1016/j.pain.2012.03.026

Weibel, S., Kunze, J., Lagoze, C., and Wolf, M. (1998). Dublin core metadata for resource discovery. *Internet Eng. Task Force RFC* 2413, 222.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 24 December 2013; accepted: 19 February 2014; published online: 11 March 2014.

Citation: Das S, McCaffrey PG, Talkington MWT, Andrews NA, Corlosquet S, Ivinson AJ and Clark T (2014) Pain Research Forum: application of scientific social media frameworks in neuroscience. Front. Neuroinform. ***8****:21. doi: 10.3389/fninf.2014.00021*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers in
NEUROINFORMATICS

GENERAL COMMENTARY
published: 26 August 2013
doi: 10.3389/fninf.2013.00017

Growing a garden of neurons

Rebekah C. Evans* and Sridevi Polavaram

Center for Neural Informatics, Krasnow Institute for Advanced Study, George Mason University, Fairfax, VA, USA
**Correspondence: rcolema2@masonlive.gmu.edu*

Edited by:
John Van Horn, University of California at Los Angeles, USA

Keywords: morphology, dendrites, growth algorithm, computational model, neural networks

A commentary on

Self-referential forces are sufficient to explain different dendritic morphologies
by Memelli, H., Torben-Nielsen, B., and Kozloski, J. (2013). Front. Neuroinform. 7:1. doi: 10.3389/fninf.2013.00001

Computational models of biologically realistic neuronal networks have advanced neuroscience in the past 20 years. With an ultimate goal of simulating a whole brain, these networks must become larger and more complex. However, a sheer massive number of neurons do not make a brain. Neurons are all different, with different kinetics, neurotransmitters, and importantly different morphologies. A network can be made by connecting copies of the same cell together, but this kind of *homogenous* network can only explain so much. Real neuronal networks are *heterogeneous* and are made up of neurons that follow both intrinsic and extrinsic cues to grow their unique dendritic arbors (Scott and Luo, 2001). In addition to *homogenous* and *heterogeneous* network models, *hybrid* network models have been implemented by creating a small heterogeneous network and replicating it to establish a larger network (Kozloski, 2011). However, modeling studies have shown that homogenous networks act differently than realistic heterogeneous ones (Mäki-Marttunen et al., 2011). Because computational neuronal networks need to grow larger to simulate complete brain regions, and because heterogeneity in a network is critical to modeling a realistic brain, algorithms for digitally generating neural morphologies are a necessary step toward this goal.

A new paper by Memelli et al. (2013) joins the field of papers providing algorithms for growing digital neurons. Their algorithm can be used to build a network consisting of millions of neurons each with a unique morphology. The current models, L-Neuron (Ascoli et al., 2001), NeuGen2.0 (Wolf et al., 2013), NetMorph (Koene et al., 2009), and CD3X (Zubler and Douglas, 2009) have made great strides in advancing the process of generating digital neurons. These models are all publicly available, and can be used to generate large networks of neurons. Recently L-Neuron was used to generate a 0.5 million cell model of the dentate gyrus (Schneider et al., 2012). Each algorithm has its own specific advantages. NetMorph has a synapse-generating algorithm, NeuGen2.0 is modular and adaptable to new data, and CD3X can isolate intrinsic and extrinsic factors of neuron development by growing the same neurons in different model environments. In combination with the parallelization of simulation software [such as NEURON (Migliore et al., 2006)], these neuron generators are laying the groundwork for enabling massive biologically realistic simulations.

Memelli et al. (2013) do not attempt to model the molecular mechanisms of dendritic growth, but instead work to make a concise, computationally efficient model that can capture the structure and variability of realistic morphologies. Their work adds two elements to this field. First, it simplifies the neural growth algorithm to contain a combination of three biologically inspired intrinsic parameters: soma-oriented tropism, dendritic self-avoidance, and membrane stiffness. The three parameters of their growth algorithm are all intrinsic to the cell itself and do not take into account any extrinsic signals that could come from other neurons or physical constraints. Each of these parameters has been previously described, but Memelli et al. are the first to combine them in one simple model. Second, their algorithm is written to be fast and massively parallel, creating the possibility for generating billions of neurons on the IBM Bluegene computer. Their algorithm can generate a neuron in less than two seconds, and when run on parallel cores is capable of generating enough neurons to simulate an entire brain region. Together, these elements fit the need to have morphological diversity within a network as well as the need to have extremely large networks.

Each of the current morphology simulators has their particular strengths. The ideal situation would be for Memelli's new algorithm to be incorporated into one of the existing ready-to-use packages. For example, the application of this algorithm within the external constraints of CX3D could help isolate the extrinsic and intrinsic aspects of dendritic arborization. When used together these simulators can help create massive-scale heterogeneous networks for computational modelers and can help investigate how dendrites actually grow.

REFERENCES

Ascoli, G. A., Krichmar, J. L., Scorcioni, R., Nasuto, S. J., and Senft, S. L. (2001). Computer generation and quantitative morphometric analysis of virtual neurons. *Anat. Embryol.* 204, 283–301. doi: 10.1007/s004290100201

Koene, R. A., Tijms, B., van Hees, P., Postma, F., de Ridder, A., Ramakers, G. J. A., et al. (2009). NETMORPH: a framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinformatics* 7, 195–210. doi: 10.1007/s12021-009-9052-3

Kozloski, J. (2011). Automated reconstruction of neural tissue and the role of large-scale simulation. *Neuroinformatics* 9, 133–142. doi: 10.1007/s12021-011-9114-1

Mäki-Marttunen, T., Aćimović, J., Nykter, M., Kesseli, J., Ruohonen, K., Yli-Harja, O., et al. (2011). Information diversity in structure and dynamics of simulated neuronal networks. *Front. Comput. Neurosci.* 5:26. doi: 10.3389/fncom.2011.00026

Memelli, H., Torben-Nielsen, B., and Kozloski, J. (2013). Self-referential forces are sufficient to explain different dendritic morphologies. *Front. Neuroinform.* 7:1. doi: 10.3389/fninf.2013.00001

Migliore, M., Cannia, C., Lytton, W. W., Markram, H., and Hines, M. L. (2006). Parallel network simulations with NEURON. *J. Comput. Neurosci.* 21, 119–129. doi: 10.1007/s10827-006-7949-5

Schneider, C. J., Bezaire, M., and Soltesz, I. (2012). Toward a full-scale computational model of the rat dentate gyrus. *Front. Neural Circuits* 6:83. doi: 10.3389/fncir.2012.00083

Scott, E. K., and Luo, L. (2001). How do dendrites take their shape. *Nat. Neurosci.* 4, 359–365. doi: 10.1038/86006

Wolf, S., Grein, S., and Queisser, G. (2013). Employing NeuGen 2.0 to automatically generate realistic morphologies of hippocampal neurons and neural networks in 3D. *Neuroinformatics* 11, 137–148. doi: 10.1007/s12021-012-9170-1

Zubler, F., and Douglas, R. (2009). A framework for modeling the growth and development of neurons and networks. *Front. Comput. Neurosci.* 3:25. doi: 10.3389/neuro.10.025.2009

Received: 02 August 2013; accepted: 02 August 2013; published online: 26 August 2013.

Citation: Evans RC and Polavaram S (2013) Growing a garden of neurons. Front. Neuroinform. **7***:17. doi: 10.3389/fninf.2013.00017*

This article was submitted to the journal Frontiers in Neuroinformatics.

GENERAL COMMENTARY
published: 05 June 2015
doi: 10.3389/fninf.2015.00014

Corrigendum: Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools

Dylan Wood *

The Mind Research Network, Albuquerque, NM, USA

Keywords: neuroinformatics, NeuroDebian, COINS, data sharing, open neuroscience

A corrigendum on

Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools.
by Wood, D., King, M., Landis, D., Courtney, W., Wang, R., Kelly, R., et al. (2014). Front. Neuroinformatics 8:71. doi: 10.3389/fninf.2014.00071

Footnotes 6 and 7:
Should both refer to the ABIDE dataset: http://fcon_1000.projects.nitrc.org/indi/abide/

Footnote 8:
Should reference the article below, in addition to the NeuroDebian website: http://neuro.debian.net.

Halchenko, Y. O., and Hanke, M. (2012). Open is not enough. Let's take the next step: an integrated, community-driven computing platform for neuroscience. *Front. Neuroinform.* **6**:22. doi: 10.3389/fninf.2012.00022. http://journal.frontiersin.org/article/10.3389/fninf.2012.00022/pdf

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

OPEN ACCESS

Edited and reviewed by:
Xi Cheng,
Lieber Institue for Brain Development,
USA

****Correspondence:***
Dylan Wood,
dwood@mrn.org

Received: *24 April 2015*
Accepted: *12 May 2015*
Published: *05 June 2015*

Citation:
Wood D (2015) Corrigendum: Harnessing modern web application technology to create intuitive and efficient data visualization and sharing tools. Front. Neuroinform. 9:14. doi: 10.3389/fninf.2015.00014

GENERAL COMMENTARY
published: 16 May 2014
doi: 10.3389/fninf.2014.00050

Corrigendum: Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness

Adrian S. Andronache*[1], *Cristina Rosazza*[1,2], *Davide Sattin*[3], *Matilde Leonardi*[3], *Ludovico D'Incerti*[1] and *Ludovico Minati*[2]

[1] *Neuroradiology Unit, Fondazione IRCCS Istituto Neurologico "Carlo Besta," Milano, Italy*
[2] *Scientific Department, Fondazione IRCCS Istituto Neurologico "Carlo Besta," Milano, Italy*
[3] *Neurology, Public Health, Disability Unit, Scientific Department, Fondazione IRCCS Istituto Neurologico "Carlo Besta," Milan, Italy*
**Correspondence: lminati@istituto-besta.it*

Edited and reviewed by:
Daniel Marcus, Washington University in St. Louis, USA

Keywords: functional MRI (fMRI), resting-state, functional connectivity, disorders of consciousness, vegetative state, minimally-conscious state, data preprocessing

A corrigendum on

Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness
by Andronache, A., Rosazza, C., Sattin, D., Leonardi, M., D'Incerti, L., Minati, L., et al. (2013) Front. Neuroinform. 7:16. doi: 10.3389/fninf.2013.00016

The authors of the article by Andronache et al. apologize that the funding acknowledgment "*The Start-up Coma Research Centre (CRC) project was also supported by the European Foundation for Biomedical Research (FERB)*" was omitted from the original publication.

REFERENCES

Andronache, A., Rosazza, C., Sattin, D., Leonardi, M., D'Incerti, L., Minati, L., et al. (2013). Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness. *Front. Neuroinform*. 7:16. doi: 10.3389/fninf.2013.00016

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 14 April 2014; accepted: 24 April 2014; published online: 16 May 2014.

Citation: Andronache AS, Rosazza C, Sattin D, Leonardi M, D'Incerti L and Minati L (2014) Corrigendum: Impact of functional MRI data preprocessing pipeline on default-mode network detectability in patients with disorders of consciousness. Front. Neuroinform. ***8****:50. doi: 10.3389/fninf.2014.00050*

This article was submitted to the journal Frontiers in Neuroinformatics.

GENERAL COMMENTARY
published: 05 September 2014
doi: 10.3389/fninf.2014.00075

Corrigendum: Software and hardware infrastructure for research in electrophysiology

Roman Mouček[1,2]*

[1] Department of Computer Science and Engineering, University of West Bohemia, Plzen, Czech Republic
[2] New Technologies for the Information Society, University of West Bohemia, Plzen, Czech Republic
*Correspondence: moucek@kiv.zcu.cz

Edited by:
Xi Cheng, Lieber Institute for Brain Development, USA

Reviewed by:
Sam Neymotin, State University of New York, Downstate Medical Center, USA
Juliana A. Knocikova, Université de Nice Sophia-Antipolis, France

Keywords: electrophysiology, event related potentials, infrastructure, neuroinformatics, workflow, portal, signal processing methods, stimulator

A corrigendum on

Software and hardware infrastructure for research in electrophysiology
by Mouček, R., Ježek, P., Vařeka, L., Řondík, T., Brůha, P., Papež, V., et al. (2014). Front. Neuroinform. 8:20. doi: 10.3389/fninf.2014.00020

This article adds Yann Le Franc as a co-author of the technology report article "Software and Hardware Infrastructure for Research in Electrophysiology," describes his individual contribution to the article, and presents changes in two paragraphs in Section 2.4, where an additional reference is also provided. Moreover, the members from the OEN group who work and cooperate on building OEN are personally acknowledged.

Co-author: Yann Le Franc

Affiliation: e-Science Data Factory S.A.S.U., Paris, France; Ludwig-Maximilians-Universität München, Planegg-Martinsried, Germany; University of Antwerp, Antwerpen, Belgium

Authors' Contributions

Yann Le Franc contributed to the **Figures 3, 4** and **8** and to the work on the Ontology for describing Experimental Neurophysiology (OEN).

Section 2.4

Old version:

The group follows the best practices for creating ontologies, for example, it cooperates with community of researchers who design and create ontologies, uses existing data formats and repositories (odML, HDF5), and reuses existing resources (terms, ontologies—NEMO, OBI). For the general description of experimental neurophysiology, the terms from ontologies NEMO and OBI are relevant. However, the set of the domain terms is still not complete in these ontologies (information stored in the EEG/ERP Portal cannot be fully described by these ontologies) and OEN will be finally an extension of OBI (e.g., the granularity of OBI for devices and related information will be extended).

New version:

The group follows the best practices for creating ontologies, for example, it cooperates with community of researchers who design and create ontologies, uses existing data formats and repositories (odML, HDF5), and reuses existing resources (terms, ontologies—NEMO, OBI). For the general description of experimental neurophysiology, the terms from ontologies NEMO and OBI are relevant. However, the set of the domain terms needed to describe the information stored in the EEG/ERP Portal is not yet complete in these ontologies. OEN aims at defining these missing terms and at term, should be used to propose an extension of OBI's neurophysiology model (e.g., the granularity of OBI for devices and related information will be extended).

Old version:

Terminologies within OEN have been primarily developed in the odML format. Subsequently, an OWL file has been constructed aided by Ontofox (Xiang et al., 2010). The current developer's version of OEN is available at https://github.com/G-Node/OEN.

New version:

The OEN device branch development is based on the odML terminology (Grewe et al., 2011), concepts gathered by the Neuroscience Information Framework (NIF) and concepts used in the EEGBase data model to describe setups and setup configurations. The gathered terms are currently mapped with the aforementioned ontologies. Subsequently, an OWL file has been constructed to contain OEN terms and the mapped terms. Existing terms in other ontologies will be imported using the MIREOT approach (Courtot et al., 2011), aided by Ontofox (Xiang et al., 2010). The current developer's version of OEN is available at https://github.com/G-Node/OEN.

ACKNOWLEDGMENTS

We thank the members of the OEN group Anita Bandrowski, Jan Grewe, Shreejoy Tripathy, and Thomas Wachtler for their work and long-term cooperation on building OEN.

REFERENCES

Courtot, M., Gibson, F., Lister, A. L., Malone, J., Schober, D., Brinkman, R. R., et al. (2011). MIREOT: the minimum information to reference an external ontology term. *J. Appl. Ontol.* 6, 23–33. doi: 10.3233/AO-2011-0087

Grewe, J., Wachtler, T., and Benda, J. (2011). A bottom-up approach to data annotation in neurophysiology. *Front. Neuroinform.* 5:16. doi: 10.3389/fninf.2011.00016

Xiang, Z., Courtot, M., Brinkman, R., Ruttenberg, A., and He, Y. (2010). OntoFox: web-based support for ontology reuse. *BMC Res. Notes* 3:175. doi: 10.1186/1756-0500-3-175

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 31 July 2014; accepted: 20 August 2014; published online: 05 September 2014.

Citation: Mouček R (2014) Corrigendum: Software and hardware infrastructure for research in electrophysiology. Front. Neuroinform. ***8****:75. doi: 10.3389/fninf.2014.00075*

This article was submitted to the journal Frontiers in Neuroinformatics.

frontiers

EPFL Innovation Park · Building I · 1015 Lausanne · Switzerland
T +41 21 510 17 00 · info@frontiersin.org · frontiersin.org