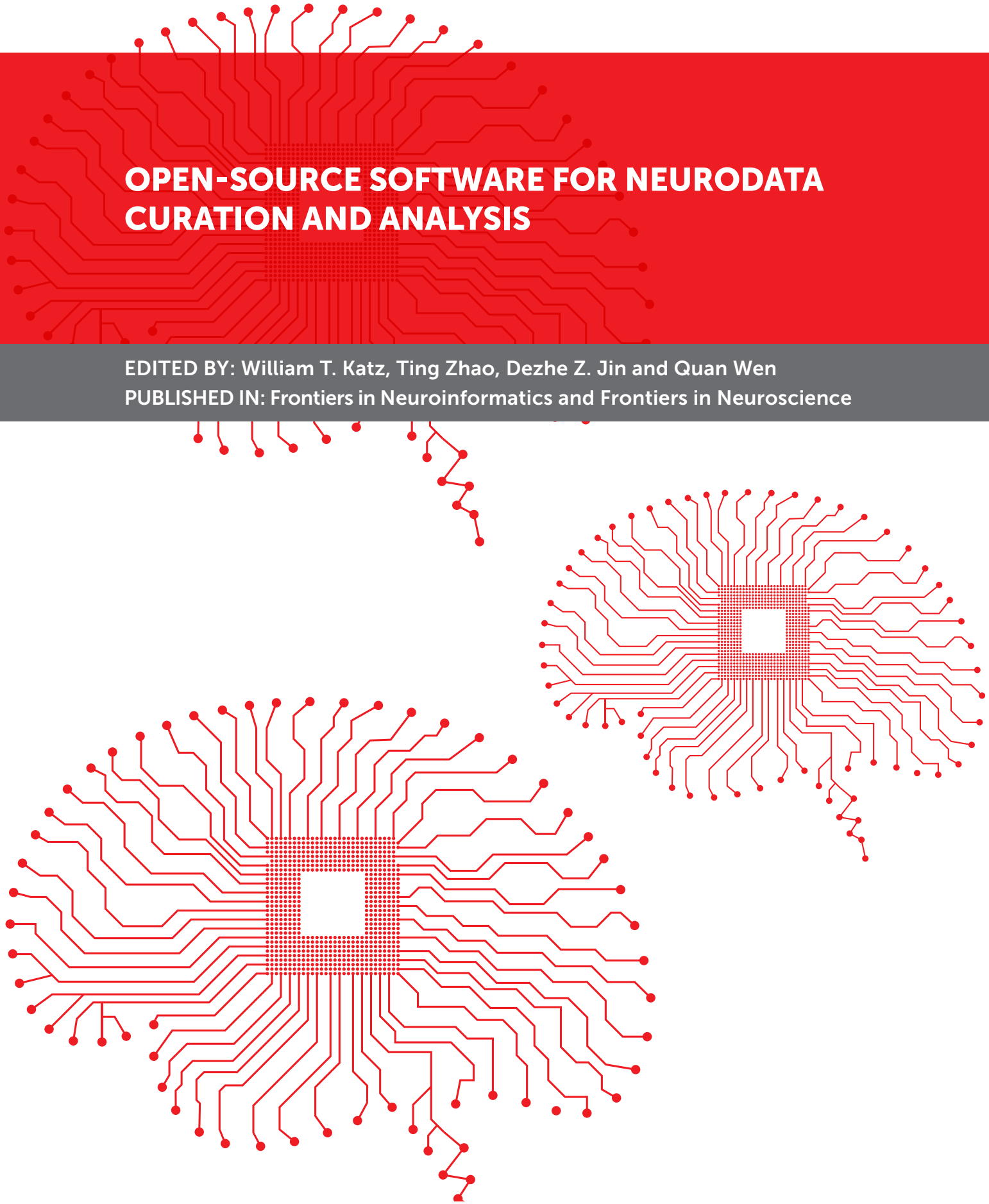
OPEN-SOURCE SOFTWARE FOR NEURODATA CURATION AND ANALYSIS

EDITED BY: William T. Katz, Ting Zhao, Dezhe Z. Jin and Quan Wen

PUBLISHED IN: Frontiers in Neuroinformatics and Frontiers in Neuroscience



frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88976-518-8
DOI 10.3389/978-2-88976-518-8

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: frontiersin.org/about/contact

OPEN-SOURCE SOFTWARE FOR NEURODATA CURATION AND ANALYSIS

Topic Editors:
William T. Katz, Janelia Research Campus, United States
Ting Zhao, Janelia Research Campus, United States
Dezhe Z. Jin, The Pennsylvania State University (PSU), United States
Quan Wen, University of Science and Technology of China, China

Citation: Katz, W. T., Zhao, T., Jin, D. Z., Wen, Q., eds. (2022). Open-Source Software for Neurodata Curation and Analysis. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88976-518-8

Table of Contents

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 18 June 2021
doi: 10.3389/fnins.2021.676220

PupilEXT: Flexible Open-Source Platform for High-Resolution Pupillometry in Vision Research

Babak Zandi[1], Moritz Lode[1], Alexander Herzog[1], Georgios Sakas[2] and Tran Quoc Khanh[1]*

[1] Laboratory of Lighting Technology, Department of Electrical Engineering and Information Technology, Technical University of Darmstadt, Darmstadt, Germany, [2] Interactive Graphic Systems, Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany

The human pupil behavior has gained increased attention due to the discovery of the intrinsically photosensitive retinal ganglion cells and the afferent pupil control path's role as a biomarker for cognitive processes. Diameter changes in the range of 10^{-2} mm are of interest, requiring reliable and characterized measurement equipment to accurately detect neurocognitive effects on the pupil. Mostly commercial solutions are used as measurement devices in pupillometry which is associated with high investments. Moreover, commercial systems rely on closed software, restricting conclusions about the used pupil-tracking algorithms. Here, we developed an open-source pupillometry platform consisting of hardware and software competitive with high-end commercial stereo eye-tracking systems. Our goal was to make a professional remote pupil measurement pipeline for laboratory conditions accessible for everyone. This work's core outcome is an integrated cross-platform (macOS, Windows and Linux) pupillometry software called PupilEXT, featuring a user-friendly graphical interface covering the relevant requirements of professional pupil response research. We offer a selection of six state-of-the-art open-source pupil detection algorithms (Starburst, Swirski, ExCuSe, ElSe, PuRe and PuReST) to perform the pupil measurement. A developed 120-fps pupillometry demo system was able to achieve a calibration accuracy of 0.003 mm and an averaged temporal pupil measurement detection accuracy of 0.0059 mm in stereo mode. The PupilEXT software has extended features in pupil detection, measurement validation, image acquisition, data acquisition, offline pupil measurement, camera calibration, stereo vision, data visualization and system independence, all combined in a single open-source interface, available at https://github.com/openPupil/Open-PupilEXT.

Keywords: pupillometry, pupil measurement, stereo camera, vision research, pupil diameter, eye tracking, open source

OPEN ACCESS

Edited by:
Ting Zhao,
Janelia Research Campus,
United States

Reviewed by:
Enkelejda Kasneci,
University of Tübingen, Germany
Cristian Rotariu,
Grigore T. Popa University of Medicine and Pharmacy, Romania

****Correspondence:***
Babak Zandi
zandi@lichttechnik.tu-darmstadt.de

Specialty section:
This article was submitted to Neural Technology, a section of the journal Frontiers in Neuroscience

Received: *04 March 2021*
Accepted: *28 April 2021*
Published: *18 June 2021*

Citation:
Zandi B, Lode M, Herzog A, Sakas G and Khanh TQ (2021) PupilEXT: Flexible Open-Source Platform for High-Resolution Pupillometry in Vision Research. Front. Neurosci. 15:676220. doi: 10.3389/fnins.2021.676220

INTRODUCTION

The pupil diameter is an essential metric in visual neuroscience, as it has a direct impact on the retinal irradiance, visual acuity and visual performance of the eye (Campbell, 1957; Campbell and Gubisch, 1966; Woodhouse, 1975; Schwiegerling, 2000). Since the early days of pupillary research (Reeves, 1918), the modeling of the pupil light response and its retinal processing path

was the main focus of investigations (Zandi and Khanh, 2021). Additionally, the pupil diameter is used as a biomarker in research disciplines such as cognitive science (Aminihajibashi et al., 2020; Cherng et al., 2020; Clewett et al., 2020; Sibley et al., 2020), circadian photoentrainment (Münch et al., 2012; Bonmati-Carrion et al., 2016; Spitschan et al., 2019; Tähkämö et al., 2019; Van Egroo et al., 2019), clinical diagnostics (Lim et al., 2016; Joyce et al., 2018; Chougule et al., 2019) or neuroscience (Schwalm and Jubal, 2017; Carle et al., 2019). Pupil changes of 0.015 to 0.5 mm are the range of interest in such studies, leading to increased resolution and robustness requirements for pupil measurement equipment. Closed commercial eye-tracking systems are common in pupil examinations, associated with high investments without offering the possibilities of validating the pupil detection's measurement accuracy. Additionally, with closed systems, it is not possible to identify the applied pupil detection algorithm, making it challenging to reproduce experiments since small inaccuracies in a range of 0.01 mm could propagate errors to the statistical evaluation of the pupil diameter. Apart from commercial solutions, there is currently a lack of an end-to-end open-source measurement platform that can be easily set up for high-precision pupillometry under laboratory conditions. Therefore, we developed a freely available hardware and software platform for pupil measurements to support the increased interest of interdisciplinary research groups in studying the pupil behavior. Our proposed platform is a comprehensive solution for performing accurate, verifiable and reproducible pupil examinations, competitive with high-end commercial stereo eye-tracking systems.

The core outcome of this work is an integrated cross-platform (macOS, Windows and Linux) pupillometry software called *PupilEXT*, featuring a user-friendly graphical interface (C++, QT), covering the relevant requirements of professional pupil behavior research (**Figure 1**). The open-source philosophy offers insight into how the pupil measurement framework performs, motivating to more transparency in collecting pupil data. We aimed to provide a plug-and-play integrated hardware and software platform, allowing interdisciplinary research groups a precise pupil behavior research without high investments. The proposed software is designed to incorporate high-resolution industrial cameras that can be run either individually or in a stereo camera arrangement. We guarantee a stable frame rate and synchronous operation of stereo cameras by using a microcontroller as an external hardware trigger. The integrated solution with hardware and software is provided in a way that even scientists with a non-technical background can reproduce the system. Users simply need to purchase industrial cameras and run the proposed *PupilEXT* software.

Inspired by the eye-tracking software *EyeRecToo* (Santini et al., 2017) from Santini et al., we offer end-users a selection of six state-of-the-art open-source pupil detection algorithms (*Starburst*, *Swirski*, *ExCuSe*, *Else*, *PuRe* and *PuReST*) to perform the pupil measurement. The system allows researchers to report the used pupil algorithm with the respective parameters since the pupil detection method itself could influence the captured data. Additionally, end-users will be able to determine the pupil diameter from externally acquired

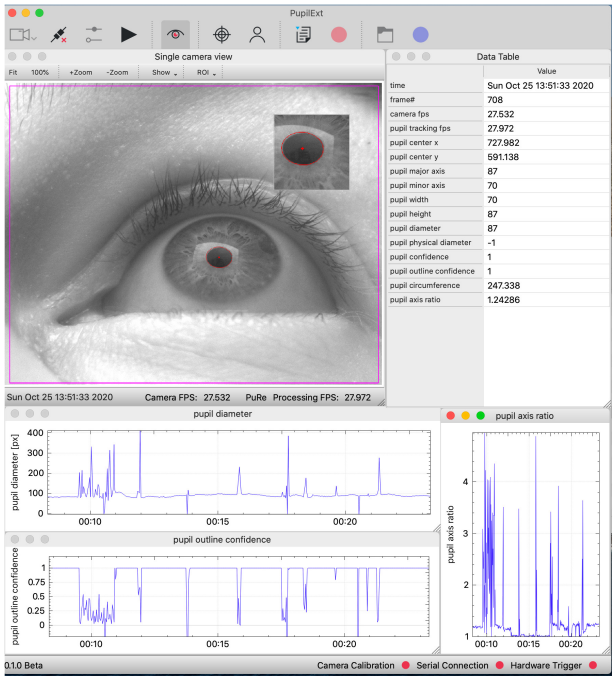


FIGURE 1 | The graphical user interface of the *PupilEXT* software during a pupil measurement with one connected industrial camera. The measured pupil values are listed in real-time in a table or can be visualized graphically. We provide a selection of six state-of-the-art pupil detection algorithms from the literature. Stereo camera systems can be connected and calibrated seamlessly to acquire the absolute pupil diameter. The accuracy of a pupil measurement or calibration can be verified by implemented routines.

image sequences through the software suite. The integrated platform is available to other research groups as an open-source project, ensuring continuous development in the future. We aimed to bridge the gap between visual neuroscience or experimental psychology and engineering sciences, making professional remote pupil measurements under laboratory conditions accessible for everyone, without suffering the features of commercial solutions.

The first section of this work deals with the scientific background of pupil behavior research and the rising popularity of this topic, from which we derive the motivation of the proposed pupil measurement platform. Based on that, the current state of pupillometry and the availability of suitable open-source frameworks are highlighted. Next, we conducted a meta-analysis of existing pupil detection algorithms from the literature intending to select and integrate appropriate algorithms in the proposed *PupilEXT* software. The functionality of the platform is covered by starting with the hardware components, consisting of cameras, microcontroller and a near-infrared (NIR) illumination. Here, we describe the possible hardware topologies with which end-users can conduct a pupil measurement or offline analysis of external captured images. In particular, we show the possibilities of validating a pupil measurement and camera calibration with the *PupilEXT* software. Finally, the performance of the system is demonstrated with an experiment concerning

the pupil light response, clarifying the provided pupil metrics for reliable data evaluation.

THE RISING POPULARITY OF PUPIL LIGHT RESPONSE RESEARCH

The human retina contains receptors with distinct photopigments, capable of transforming light quanta of different wavelengths λ into frequency-coded action potentials with information on color and brightness features from a visual stimulus. Photoreceptors in the retina are classified according to their broad spectral sensitivity in the visible spectrum range and respective peak response λ_{Peak}. In the photopic adapted eye, the retinal image-forming pathway is mainly controlled by the short-wavelength (S, λ_{Peak} 420 nm), medium-wavelength (M, λ_{Peak} 535 nm) and long-wavelength (L, λ_{Peak} 565 nm) sensitive cones (Stockman and Sharpe, 2000; Solomon and Lennie, 2007; Lucas et al., 2014). At scotopic and mesopic light conditions, the more sensitive rods (λ_{Peak} 498 nm) dominate the vision. Both cones and rods transmit, depending on the adaptation state of the eye, integrated signals in different stages through ganglion cells to the visual cortex of the brain (Van Meeteren, 1978; Smith et al., 2008; Jennings and Martinovic, 2014). In 1924, the International Commission on Illumination (CIE) introduced the photopic luminous efficiency function $V(\lambda)$ to estimate the visual effectiveness of light spectra for humans (Bodmann, 1992; Sharpe et al., 2005; Sagawa, 2006).

A standard value in estimating the human brightness perception is the luminance L given in cd/m^2, which is a $V(\lambda)$ weighted photometric quantity (Berman et al., 1990; Lennie et al., 1993; Withouck et al., 2013). The luminance is merely a first approximation of the brightness perception, as only the additive contribution of L- and M-cones to the image-forming pathway is managed by $V(\lambda)$ (CIE, 2011; Besenecker and Bullough, 2017; Hermans et al., 2018; Zandi et al., 2021). Since 1926, about eight pupil models were proposed that integrated the luminance as a main dependent parameter, assuming that the afferent pupil control pathway can be described by a $V(\lambda)$ weighted quantity (Holladay, 1926; Crawford, 1936; Moon and Spencer, 1944; de Groot and Gebhard, 1952; Stanley and Davies, 1995; Blackie and Howland, 1999; Barten, 1999; Watson and Yellott, 2012; Zandi et al., 2020).

The discovery of a new type of receptors in the outer retina called intrinsically photosensitive retinal ganglion cells (ipRGCs) was a turning point of vision science (Provencio et al., 1998, 2000; Gooley et al., 2001; Berson et al., 2002; Hattar, 2002; Mure, 2021), which has led to a rethinking of classical retinal processing models. This subset of ganglion cells are part of the non-image-forming mechanism of the eye because of their projection to regions of the suprachiasmatic nucleus (SCN) and olivary pretectal nucleus (OPN) (Ruby et al., 2002; Berson, 2003; Hattar et al., 2003; Do et al., 2009; Ecker et al., 2010; Allen et al., 2019; Do, 2019). As a result, the ipRGCs can modulate the circadian rhythm (Freedman, 1999; Brainard et al., 2001; Thapan et al., 2001; Rea and Figueiro, 2018; Truong et al., 2020) and pupil light response (Lucas et al., 2001, 2020; Gamlin et al., 2007; Young and Kimura, 2008; Barrionuevo et al., 2018; Murray et al., 2018) via a processing path that works independently of the classical image-forming pathway (Hattar et al., 2006; Güler et al., 2008; Schmidt et al., 2014; Spitschan, 2019a). Recent studies showed that the pupil light response cannot be described by the $V(\lambda)$ weighted luminance alone, making a revision of classical pupil models necessary (Zandi et al., 2018, 2020; Spitschan, 2019b; Zele et al., 2019). Therefore, one key topic in pupillary research is the development of a valid empirical model (Zandi et al., 2020), providing a spectral and time-variant function with dynamic receptor weighting to predict the temporal aperture across individuals (Rao et al., 2017; Zandi and Khanh, 2021). When using stimulus spectra along the Planckian locus for triggering the pupil light response, it is essential in measurements that amplitudes in the range of 0.1 to 0.4 mm are captured accurately to specify intrasubject variability (Kobashi et al., 2012) in a pupil model. However, a special requirement for pupil measurements arises when the pupil is used as a biomarker for quantifying the cognitive state (Morad et al., 2000; Merritt et al., 2004; Murphy et al., 2014; Ostrin et al., 2017; Tkacz-Domb and Yeshurun, 2018; Hu et al., 2019; Van Egroo et al., 2019; de Winter et al., 2021; Van der Stoep et al., 2021) or clinical symptoms of diseases (Hreidarsson, 1982; Maclean and Dhillon, 1993; Connelly et al., 2014; Lim et al., 2016; Granholm et al., 2017; Wildemeersch et al., 2018; Chougule et al., 2019). Cognitive processes such as memory load, arousal, circadian status, or sleepiness have a transient impact (Watson and Yellott, 2012) on the pupil diameter with aperture changes of 0.015 to 0.53 mm (Beatty and Wagoner, 1978; Beatty, 1982; Schluroff et al., 1986; Jepma and Nieuwenhuis, 2011; Pedrotti et al., 2014; Bombeke et al., 2016; Tsukahara et al., 2016; Winn et al., 2018), making the reproducibility of such effects difficult if the accuracy of the measurement equipment has not been sufficiently validated.

Today, the pupil behavior has become an interdisciplinary field of research (La Morgia et al., 2018; Schneider et al., 2020; Joshi, 2021; Pinheiro and da Costa, 2021) in which the number of involved scientists rises, as the trend of the number of publications with the keywords "pupil diameter" or "pupillometry" reveals (**Figure 2**). The renewed attention to the temporal pupil aperture (Binda and Gamlin, 2017), its application in clinical diagnostics (Granholm et al., 2017; Joyce et al., 2018; Chougule et al., 2019; Kercher et al., 2020; Tabashum et al., 2021) and increasing popularity of chromatic pupillometry (Rukmini et al., 2017; Crippa et al., 2018) topics requires additional efforts in terms of standardization and provision of consistent tools, contributing to comparability in measurement and pre-processing methodologies. For instance, one key point of standardization is the prevention of artificially induced changes to raw data by the used tools, as in cognitive or vision-related pupillary research small diameter margins are of interest. The main methodology factors that could influence the research results or reliability of pupil behavior studies are as follows:

(1) Number and depth of described experimental metrics when reporting the results concerning the stimulus modality or pre-conditioning state of the subjects.

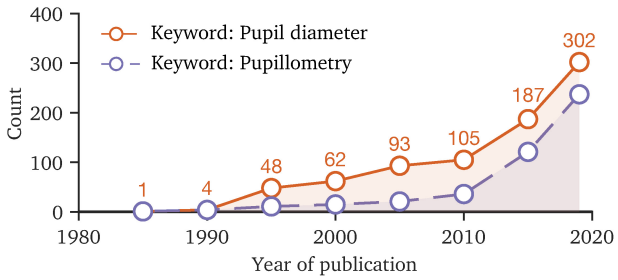


FIGURE 2 | The number of publications with the keywords "pupil diameter" and "pupillometry" since 1985 to 2019, based on the Web of Science database. The rising count of publications in recent years indicates that the topic of pupil behavior is becoming more important. Due to the interdisciplinary field of research, standardization of measurement methodology and data processing is favorable, making study results comparable.

(2) The used pre-processing method to smooth out and clean the measured pupil raw data.
(3) The used measurement hardware and software framework in collecting pupil data.

In order to minimize the influencing factors, there are actions in the research community to provide the essential tools for pupil research to lower the barrier of entering the topic and ensuring the comparability of future research. A major step in this direction was the work "Standards in Pupillography" by Kelbsch et al., which summarized the current knowledge on pupil behavior and defined recommendations to be considered by author groups when reporting pupil study results (Kelbsch et al., 2019). The standardization approach mainly dealt with the minimal set of metrics that authors need to specify in published research, allowing third parties to reproduce experiments when necessary. Regarding the topic of data pre-processing, the focus is on which methods should be used to detect and remove artificially induced pupil changes, caused by eye blinks and fast gaze jumps during pupil recording sessions. Ranging from catching artifacts to smoothing out the measured raw data, a large number of software libraries and guidelines exist that can assist researchers in carrying out such tasks (Pedrotti et al., 2011; Canver et al., 2014; Lemercier et al., 2014; Attard-Johnson et al., 2019; Kret and Sjak-Shie, 2019; van Rij et al., 2019).

The research area of pupil behavior benefits from the interdisciplinarity of the research groups, which is promoted by the provision of tools and predefined standardized methodologies. However, the pupillometry technique itself is a significant hurdle, since there are no standardized requirements or reliable end-to-end open-source systems for recording pupil data in high-precision experiments under laboratory conditions.

THE ISSUE OF PUPILLOMETRY

Typically, a pupil measurement can be performed manually by using a double-pinhole pupillometer (Holladay, 1926) or photographs with a reference object (Crawford, 1936) or through an integrated eye-tracking system. A higher proportion of pupil behavior studies is conducted by using an eye-tracking system, as identifying the pupil region is often a necessary step before estimating the gaze position (Lee et al., 2012). Commercial eye trackers from Tobii Pro, Smart Eye Pro or Eyelink are common solutions, which are easy to set up and usable without a technical background but cost approximately between 5,000 and 40,000 euros (Hosp et al., 2020; Manuri et al., 2020). Purchasing a set of high-resolution professional industrial cameras costs about 200 to 600 euros, with which an optical accuracy of 0.01 mm/px or more could be achieved. Thus, the price gap from commercial products results from the integrated software and license fees.

Commercial systems rely on closed software, restricting thereby conclusions about the used pupil-tracking algorithms, which is essential for the reproducibility. Additionally, based on the authors' best knowledge, there is no commercial eye-tracking system that states the accuracy of their measured pupil diameter in the datasheet nor is a manual validation possible, as their solutions' primarily focus is on gaze tracking. Especially in studies where pupil diameter effects are in a range of 10^{-2} mm, a validation of the system's pupil measurement accuracy through a reference object is desirable.

The open-source head-mounted eye tracker project by *Pupil Labs* (Kassner et al., 2014) is an alternative to fully commercialized solutions, allowing free head movements and experiments in natural environments where a classic remote eye-tracking set-up is not possible. However, we do not recommend this system for precise pupil measurement applications, due to the cameras' positions which are highly off-axis, causing pupil foreshortening errors (Hayes and Petrov, 2016). Additionally, the absolute pupil diameter is calculated indirectly by a method from which conversion accuracy is not yet fully validated for pupil measurements. Therefore, the solution provided by *Pupil Labs* is more suitable for experiments in which only the relative pupil diameter is of interest.

Remote tracking systems, positioned on the optical axis of the eye, are better suited for reliable pupil measurements. Various published approaches provide isolated components to build a custom remote stereo camera system (Hiley et al., 2006; Long et al., 2007; Kumar et al., 2009; San Agustin et al., 2010), which is not always feasible for interdisciplinary research groups, leading to a preference for commercial solutions. However, a groundbreaking project called *EyeRecToo* by Santini et al. (2017) has taken the first steps in establishing the idea of a competitive open eye-tracking software suite, which even has the option of choosing between different state-of-the-art pupil detection algorithms. Unfortunately, the software is mainly designed for head-mounted eye trackers or webcams and the use-cases are not targeted for the experimental pipeline of pupil research under laboratory conditions. For instance, a stereo camera arrangement with extrinsic calibration and the subsequent validation of a camera's accuracy is not possible, to our best knowledge. Additionally, the software does not offer the option for evaluating external captured images from a stereo or mono camera system.

The success of the *Pupil Labs* project shows that end-users wish to have a fully integrated system consisting of hardware and software, packed with the functionalities of a professional

commercial solution. Thus, in developing our proposed platform, we have focused not only on the functionalities and requirements of pupil researchers but also on the end-user's experience, which should provide an easy way to build and run a pupil measurement system.

CHOOSING PUPIL DETECTION ALGORITHMS FOR PupilEXT

The main application for an eye-tracking system is the estimation of a subject's gaze location, which usually needs to recognize the pupil contour and its center position. Due to the high contrast between the sclera and the pupil region in a digital image, the recognition of the pupil is in principle possible through a combination of thresholding, edge detection and morphological operations (Goñi et al., 2004; Keil et al., 2010; Topal et al., 2017). State-of-the-art pupil detection approaches have additional steps in the image processing pipeline, ensuring a more robust contour fit while having a high and accurate detection rate. Under laboratory conditions, eye images are mainly captured using a NIR light source to avoid cornea reflections of the ambient environment, leading to optimized pupil detection. However, accurate pupil detection is an essential step in eye-tracking systems since a flawed edge detection could have an impact on the performance of an eye tracker (Santini et al., 2018a). Therefore, pupil detection methods intended for eye-tracking systems can also be used for pupil measurement, if an algorithm features the detection of aperture sizes.

There are three different illumination set-ups proposed for capturing a series of eye images that need to be in line with the used pupil detection algorithm (Li et al., 2005). In the bright-pupil method, a NIR-light source is placed close to the optical axis of a camera, resulting in a positive contrast between the iris and pupil region (Hutchinson et al., 1989). Due to the retinal reflection of the illumination back to the camera, the pupil region appears brighter than the iris and sclera itself (Li et al., 2005). In the dark-pupil method, the light source is placed off-axis to the camera. Thus, the pupil appears as a dark spot surrounded by the brighter iris (negative contrast). A third method called the image-difference technique leverages the image difference between dark- and bright-pupil to extract the pupil's contour. For this, one NIR illumination should be positioned close to the camera's optical axis (NIR 1) and a second one off-axis (NIR 2). By synchronizing the illuminations' flashing interval with the sampling rate of a camera, one positive contrast image can be captured in a first frame (NIR 1, ON; NIR 2, OFF) and a second frame with negative contrast (NIR 1, OFF; NIR 2, ON). This approach can lead to a more robust pupil detection but has the drawback that more effort has to be invested in the illumination. Furthermore, two frames are needed for each captured pupil size value, reducing the overall sampling rate. The recent work of Ebisawa (1994, 2004), Morimoto et al. (2002), and Hiley et al. (2006) used this image-difference technique.

However, the core of a pupil measurement system is the algorithm that is used to determine the pupil diameter. Recently published works developed state-of-the-art approaches that can be applied in our proposed software *PupilEXT*. Similar to the work of Topal et al. (2017), we conducted a meta-analysis of 35 published pupil detection methods (**Table 1**) to evaluate and select suitable algorithms for our proposed measurement platform.

The potential algorithms need to estimate the pupil size, as this is the main focus of this work. From the 35 evaluated algorithms, we can rule out 11 approaches since they are not able to output the pupil size (**Table 1**). We decided to consider only algorithms designed for dark-pupil detection, serving to more freedom in setting up the position of the NIR light source. Another criterion for the selection was the availability of the implementation since we started from the working hypothesis that published procedures with existing programming code are ready for practical applications. Since our graphical user interface (GUI) should offer real-time pupil detection, only C++-implemented approaches were of interest.

Based on these criteria and taking the algorithms' recency into account, we selected a total of six pupil detection approaches for *PupilEXT*. First, we decided to use the robust *Starburst* algorithm by Li et al. (2005), which was considered as a standard approach in pupil detection for a long time, implemented in several works throughout the years. Furthermore, we added the algorithm by Świrski et al. (2012), *ExCuSe* by Fuhl et al. (2015), *ElSe* by Fuhl et al. (2016a), *PuReST* by Santini et al. (2018b) and *PuRe* by Santini et al. (2018a). The algorithms *ElSe*, *ExCuSe*, *PuRe* and *PuReST* are licensed for non-commercial use only. The pupil detection algorithm from Swirski et al. is licensed under MIT, and the *Starburst* algorithm under GNU GPL. More details about the licensing terms of the detection algorithms can be found on the project page of *PupilEXT*[1].

We did not select pupil detection approaches based on neural networks (Mazziotti et al., 2021). Models such as *DeepEye* (Vera-Olmos et al., 2018) and *PupilNet* (Fuhl et al., 2016b, 2017) reveal promising results, but their computational complexity is still too high for real-time pupil measurement applications without special hardware.

The user has the option to choose between these state-of-the-art algorithms for pupil measurement in the proposed *PupilEXT* platform. Additionally, the algorithms' parameter can be checked and adjusted in the user interface to increase the software-based measurement accuracy, if necessary. By default, the *PuRe* algorithm is selected because it is considered as a top performer and the number of parameters are relatively user-friendly, making it to a generalized procedure for different measurement settings (Santini et al., 2018a,b). While the algorithms are solely based on recent publications from various author groups, the interested readership is referred to the original works of the respective pupil detection methods or works that already reviewed the algorithms (Topal et al., 2017; Manuri et al., 2020).

[1]https://github.com/openPupil/Open-PupilEXT

TABLE 1 | Comparison of the pupil detection algorithms identified in the literature.

Algorithm	Approach basis	Downscaling	Bright/dark pupil	Thresholding	Ellipse fitting	Center of mass	Temporal information	Runtime in ms	Pupil size output	Blink detection	Confidence measure	Implementation available	Pupil size evaluation
Ebisawa, 1994	Image-diff.		⊙	●		●							
Zhu et al., 1999	Curvature		■	●	LSM	●			●				
Morimoto et al., 2000	Image-diff.		⊙	●		●	●	67					
Pérez et al., 2003	Threshold		■	●		●		40	●				
Lin et al., 2003	Edge		■		LSM	●		166	●				
Goñi et al., 2004	Threshold		□	●		●	●	33					
Ebisawa, 2004	Image-diff.		⊙	*		●	●	20		●			
Starburst Li et al., 2005	Rays		■	*	RANSAC		●	100 (3)	●	●	○	●	
Hiley et al., 2006	Image-diff.		■	●		●		12 (2)					
Long et al., 2007	Threshold	○	■	●		●(1)		6.67					
Dey and Samanta, 2007	Threshold	●	■	*	Circle			127	●				●
San Agustin et al., 2010	Threshold		■	●	RANSAC	●			●			●	
Kumar et al., 2009	Edge		■	*	LSM	●			●	●			
Keil et al., 2010	Threshold		■	●		●		60					
Lin et al., 2010	Threshold	○	■	●	●	●(1)			●	○			
Lanatà et al., 2011	Threshold		■	●	LSM				●				
Świrski et al., 2012	Threshold		■	*	RANSAC	●		3.77	●			●	●
Schwarz et al., 2012	Threshold		■	●					●			●	
Świrski and Dodgson, 2013	3D model		■	*	RANSAC	●			●			●	●
Kassner et al., 2014	Edge		■	●	LSM			45 (3)	●	●	●	●	●
Chen and Epps, 2014	Threshold		■	*	LSM			60 (2)	●	●	○		●
ExCuSe (Fuhl et al., 2015)	Edge	●	■	●	LSM			7	●	●		●	
SET (Javadi et al., 2015)	Threshold		■	●	●	●		100	●			●	
ElSe (Fuhl et al., 2016a)	Edge	●	■	*	LSM	●		7	●	●	○	●	
PupilNet (Fuhl et al., 2016b)	CNN	○	■										
APPD (Topal et al., 2017)	Curvature		■		MSM			5.37	●	●		○	●
PuRe (Santini et al., 2018a)	Edge	●	■		LSM			5.17	●	●	●	●	
PuReST (Santini et al., 2018b)	Edge	●	■	*	LSM		●	1.88	●	●	●	●	
Li et al., 2018	Edge		■		LSM				●				
DeepEye (Vera-Olmos et al., 2018)	CNN		■	*		●		33 (4)				●	
FREDA (Martinikorena et al., 2018)	Image-diff.		■		●			63 (2)	○			●	
CBF (Fuhl et al., 2018b)	Feature-class.	●	■					6.8				●	

(Continued)

TABLE 1 | Continued

Algorithm	Approach basis	Downscaling	Bright/dark pupil	Thresholding	Ellipse fitting	Center of mass	Temporal information	Runtime in ms	Pupil size output	Blink detection	Confidence measure	Implementation available	Pupil size evaluation
BORE (Fuhl et al., 2018a)	Edge		■		●			15	●			●	
DeepVOG (Yiu et al., 2019)	CNN	●	■	●	●			17 (4)	●	●	●	●	●
Eivazi et al., 2019	CNN	●	■					8 (4)	●				

*We classified the algorithms by their applied technique and properties. This meta-analysis is inspired by the approach of Topal et al. (2017). Only a small portion of the proposed algorithms was evaluated through the detected pupil size, as stated in the last column. Typically, the authors assessed the pupil center's accuracy, as eye tracking was the main application of most algorithms. The algorithms in the gray marked rows are available in our proposed pupil measurements software PupilExt. ○ only partial, ■ dark pupil, □ bright pupil, ⊙ bright and dark pupil, * adaptive thresholding, ● yes. (1) Symmetric center of mass, (2) MATLAB implementation, (3) Python implementation, and (4) graphics processing unit (GPU) aided.*

HARDWARE SET-UP OF THE CAMERA SYSTEM

We linked the *PupilEXT* software with a specific camera brand (Basler) to provide a comprehensive platform for pupillometry. In this way, we allow a plug-and-play usage of the proposed system since the software is adapted to the hardware. The Pylon SDK is used to interface the cameras with the measurement software *PupilEXT*. Thus, any Basler branded industrial camera is integrable into the pupillometry platform. We explicitly do not support consumer webcams since *PupilEXT* is intended for reliable and accurate research applications. Generally, live or post-acquisition pupil measurements are supported through different measurement configurations (**Figure 3**).

Two cameras are needed for the stereo camera arrangement to detect the absolute pupil diameter directly (**Figure 3A**). One essential factor in the processing accuracy of such a configuration is the synchronization level between the cameras. Therefore, we synchronized the cameras through an external hardware trigger, leading to a stable system comparable with a professional manufactured commercial solution. Such a hardware trigger is needed to acquire images from both cameras simultaneously. In low-budget systems, the image acquisition is usually made by a software trigger that cannot guarantee synchronized image acquisitions, leading to reduced measurement accuracy. In our proposed system, the trigger signal is generated through a microcontroller, which is automatically controlled by *PupilEXT*. Additionally, we support pupil measurements with a single camera (**Figures 3B–D**). Here, the integration of a microcontroller for triggering an image acquisition is optional (**Figure 3B**). However, by including a microcontroller in the one-camera set-up, the duration of a recording session can be set. Note that when using a single camera, the pupil diameter is measured in pixels. Through an extra recording trial with a reference object, the pixel values can be manually converted to millimeters. If cameras are connected to *PupilEXT*, a real-time pupil measurement with one of the six pupil detection algorithms can be carried out. Furthermore, we support the option of recording images without pupil detection. In this way, it is possible to analyze the images in a post-acquisition mode without connected cameras (**Figure 3E**). In such an offline mode, image sequences from externally recorded cameras can also be loaded, making it possible to leverage the software on already existing pupil image datasets.

We recommend a NIR illumination unit to avoid corneal light reflections in the eye from the visible spectrum, which could impact the accuracy of pupil detection. For this, a NIR bandpass filter should be mounted in front of the camera's lens. The advantage of a NIR-based measurement is that the image quality does not suffer in pupil light response experiments. Both the source code of the microcontroller for generating the hardware trigger and the respective NIR circuit board design (**Figure 3D**) are provided together with the *PupilEXT* software, allowing to set up the system effortlessly. The following subsections deal with the

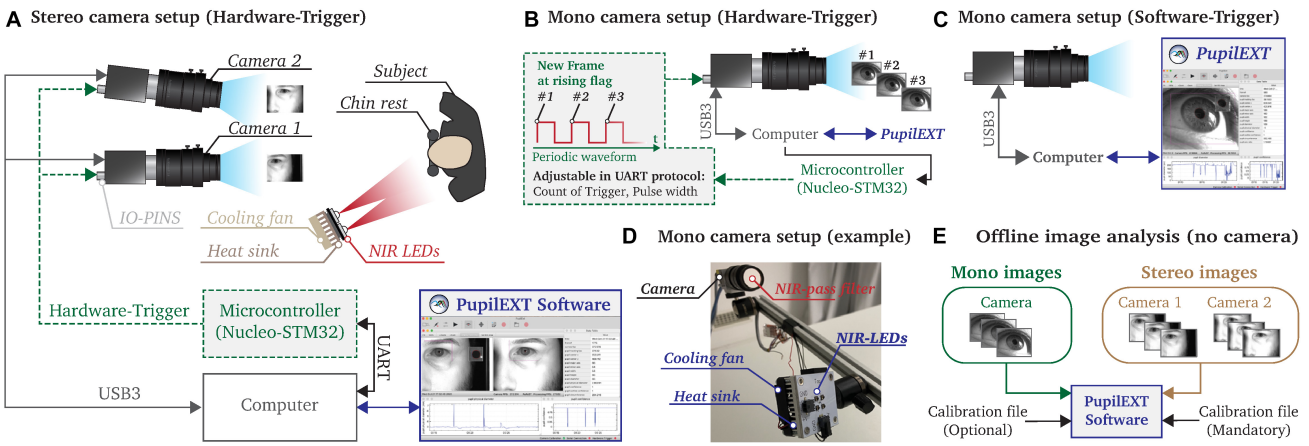


FIGURE 3 | Illustration of the possible measurement configurations that can be realized with the *PupilExt* software. **(A)** In stereo vision mode, two cameras are connected to the computer via USB3.0. A microcontroller is connected to the IO-pins of the camera, which triggers a synchronized image acquisition on both cameras. We recommend using a Nucleo-STM32 microcontroller since the provided source code can be used to flash the electronic. We implemented a UART communication protocol in the microcontroller so that *PupilExt* can automatically control and synchronize the stereo camera system via the connected hardware, via the electronics. **(B)** In mono camera vision mode, it is possible to control the image acquisition by an external hardware trigger, which has the advantage that the recording time can be set accurately. When capturing multi-image sequences, the hardware trigger consists of a square wave signal in which each rising edge triggers an image acquisition. **(C)** The use of a microcontroller is optional when connecting a single camera. Without the use of a microcontroller, a software trigger is used for image acquisition. **(D)** Prototype of a pupillometry set-up with a single camera and respective near-infrared (NIR) illumination unit. **(E)** The software *PupilExt* can be used without connected cameras in an offline mode to detect the pupil diameter from externally captured images.

different operational configurations of the platform (**Figure 3**) and the needed hardware elements in more detail, ensuring the reproducibility of the measurement platform.

Camera Set-Up

We built a prototype consisting of two Basler acA2040-120um cameras with 50-mm lenses to validate the pupillometry platform in a sample study. The cameras operated in stereo vision mode to measure the absolute pupil diameter. The cameras support a resolution of 2,048 px × 1,536 px with a maximal frame rate of 120 fps. We positioned the system in front of an observer at a working distance of 700 mm, with a baseline distance between the cameras of 75 mm in which the secondary camera has an angle of 8° to the main camera (**Figure 3A**). A NIR illumination unit, consisting of four LEDs with a peak wavelength of 850 nm (SFH-4715AS), is placed near the subject's head without obstructing the view of the cameras. Furthermore, the camera lenses are equipped with a high-pass infrared filter (Schneider IF 092 SH) with a transmission range of 747 to 2,000 nm, which should reduce artifacts from the ambient illumination.

The cameras are connected through their USB 3.0 interface with the computer for data transmission. Additionally, the IO-Pin connector of the cameras is used to adjust the timing, execution and synchronization of the image capturing. A microcontroller (Nucleo STM32F767ZI) is integrated into the pupillometry platform, controlling the cameras' capturing interval through a shared digital signal.

For this, the microcontroller transmits a periodic square waveform modulated signal with a voltage amplitude of 3.3 V. Each rising edge of the signal triggers an image (**Figure 3B**). The frequency and duration of the square wave signal are adjustable through *PupilEXT*, affecting the frame rate and recording time of the camera. While the use of a microcontroller is obligatory when shooting stereo vision, it can be used optionally in the single-camera set-up (**Figures 3B,C**). Before an absolute pupil measurement can be carried out in stereo vision mode, extrinsic and intrinsic calibrations of the cameras need to be performed in *PupilEXT*.

Embedded Hardware Trigger

In stereo vision mode, the microcontroller must be connected to the computer so that *PupilEXT* can communicate with the embedded electronic via UART. We have implemented a simple text-based protocol in the microcontroller, for starting and stopping the trigger signal. Control commands can be dispatched via the graphical interface in *PupilEXT* or manually through a serial port terminal application like *CoolTerm* or *HTerm*. If the provided embedded microcontroller source code is not used, users can easily implement the protocol themselves in their preferred microcontroller brand.

To start a trigger signal, the parameters *COUNT_OF_TRIGGER* and *TIME_TRIGGER_ON* must be set in the protocol. The parameter *COUNT_OF_TRIGGER* indicates how many rising flags should be transmitted. The parameter *TIME_TRIGGER_ON* sets the pulse width in microseconds, which is used to set the sampling rate of the camera. Both parameters are set with the string command < *TxCOUNT_OF_TRIGGERxTIME_TRIGGER_ON* > via the UART interface of the microcontroller. The "x" term is used as a separator between the parameters. For instance, if a trigger signal should be used for capturing a total of 100 images with a rate of 10 ms, the protocol would

correspond to $< Tx100x5000 >$. A detailed introduction of how to flash and install the embedded electronic is provided on the project's webpage.

THE CROSS-PLATFORM SOFTWARE SUITE

The core of the pupillometry platform consists of the software *PupilEXT*, structured and implemented based on the requirements of scientifically oriented pupil behavior research. Although pupil measurements can be performed with commercial eye-tracking solutions, the closed system design blocks the transparency of used pupil detection algorithm and the determination of its pupil measurement accuracy. Moreover, such commercial systems are not fully intended for absolute pupil measurements. With *PupilEXT*, we offer not only a free alternative to commercial solutions but also extended features in the topics of pupil detection, measurement resolution, data acquisition, image acquisition, offline measurement, camera calibration, stereo vision, data visualization and system independence, all combined in a single open-source interface.

It is possible to choose between the six discussed pupil algorithms (*Starburst*, *Swirski*, *ExCuSe*, *ElSe*, *PuRe* and *PuReST*) and to freely adjust their processing parameters and to optimize the pupil contour's detection accuracy. Additionally, the parameters of a pupil detection method can be reported, leading to an increase in the reproducibility of pupil examinations. We have integrated the pupil detection methods into one unified framework by using a standard pupil detection interface (**Figure 4**).

For this, the *PupilDetectionMethod* interface is adapted from the *EyeRecToo* eye-tracking software (Santini et al., 2017), which employs an interface to integrate multiple pupil detection algorithms. It defines a set of abstract methods like *run* and *hasConfidence*, which are concretized through the specific algorithm implementation (Santini et al., 2017). The *run* method defines the respective pupil detection algorithm that returns a detected pupil from an image. Through *hasConfidence*, we verify the availability of a confidence measure from a respective algorithm. The interface provides a general confidence measure that can be used if an algorithm does not provide its confidence measure (Santini et al., 2018a). An additional component that is adapted from *EyeRecToo* (Santini et al., 2017) is the *Pupil* class, which aggregates all data of a detected pupil and its fitted ellipse into one class. A simplified UML diagram of the adapted structure is illustrated in **Figure 4**.

In *PupilEXT*, the camera frame rate is adjustable up to 120 Hz. Pupil measurement data are stored in a comma-separated value (CSV) file containing the pupil diameter, confidence measure and ellipse parameters. Besides recording real-time pupil data, the software features storage of raw images for later pupil evaluation. A comprehensive stereo and mono calibration procedure within the software guarantees an accurate and validatable measurement pipeline. The unique feature is the integration of professional industrial cameras with stereo vision capabilities, dedicated to absolute pupil diameter measurements. Metrics are visualized in real-time during pupil measurements, providing an *ad-hoc* evaluation of metrics.

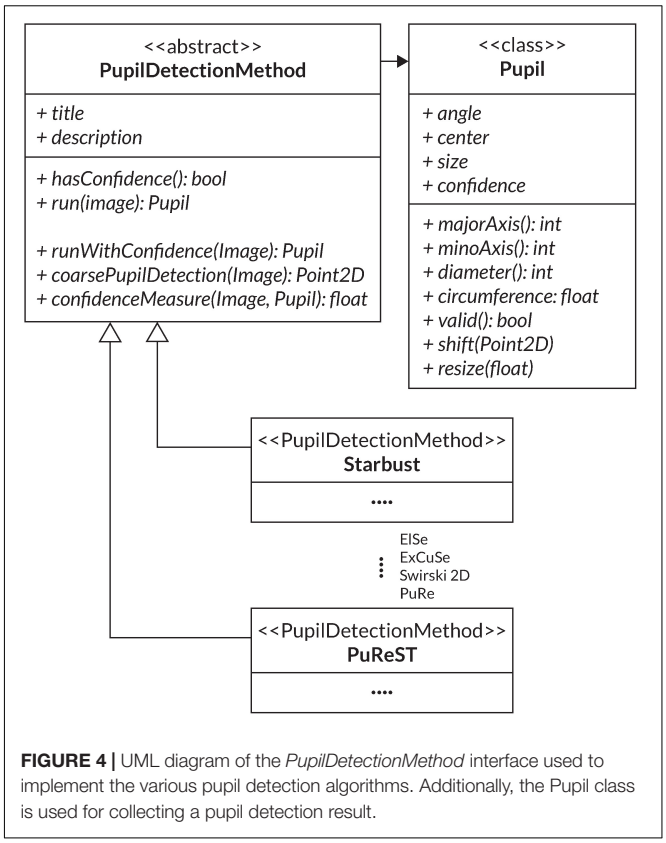


FIGURE 4 | UML diagram of the *PupilDetectionMethod* interface used to implement the various pupil detection algorithms. Additionally, the Pupil class is used for collecting a pupil detection result.

Camera Interface

Before *PupilEXT* can perform a remote pupil detection, images must be grabbed from the camera(s). We access the Basler cameras with their USB 3.0 interface using a manufacturer-provided programming library called *Pylon*. Through the library, we configure both the camera preferences and activate an image capturing trigger for passing to the image processing pipeline. We distinguish between two image acquisition modes of a camera. With a software trigger, the camera acquisition is controlled over the *Pylon* library interface to record images at a specified frame rate continuously. In the single-camera mode, commonly, the software trigger is used, and the hardware trigger is optional. The hardware trigger is mainly implemented for the stereo vision mode, in which two cameras synchronously capture images upon a receiving a signal flag on an IO-pin. In stereo camera set-ups, the integration of the hardware trigger is obligatory. In such set-ups, a software trigger cannot guarantee that both cameras capture an image at the same time, affecting the performance of a stereo system. Connection establishment and message transmission to the microcontroller is accomplished via a serial port. The microcontroller configuration includes the settings for a camera frame rate as well as the duration of the recording.

To integrate the camera(s) in *PupilEXT*, a *Camera* interface was created, defining a set of functions for all camera types (**Figure 5**). Three types of cameras are differentiated: a single

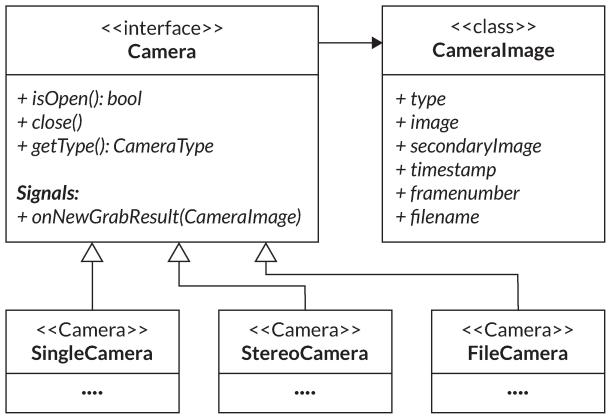


FIGURE 5 | UML diagram of the Camera interface and its implementations modeling for different types of cameras in the *PupilEXT* software. The *CameraImage* class is used to represent resulting images and their corresponding metadata.

camera, a file camera and a stereo camera consisting of the main and secondary cameras (**Figure 5**). The file camera can be viewed as a camera emulation used in offline pupil detection sessions from previously recorded images retrieved from disk storage. However, by emulating the playback of images as a camera object, it can be integrated seamlessly into existing functions of *PupilEXT*. For the representation of the camera image, the *CameraImage* class is defined (**Figure 4**). The image distribution in the *PupilEXT* software from the camera(s) is organized with an internal event handler function. For this, the *Pylon* camera library provides an interface that is called every time a corresponding camera acquires a new image. However, for a stereo camera set-up, an image recording consists of two corresponding images that will be delivered by two separate function calls.

The initial approach was to leverage a camera internal timestamp to associate the two corresponding images. However, matching the two cameras, internal timestamps of corresponding images led to a buggy image rectification. Therefore, it was necessary to find a more reliable approach. Besides the camera(s) internal timestamp, additional metadata such internal frame count is provided by the *Pylon* API. As long as both cameras start the acquisition simultaneously, the frame counts match. This approach ensures a fixed and reliable order of stereo image acquisitions processed by *PupilEXT*.

Image Recording and Reading for Offline Analysis

For retrospective detection of the pupil diameter, raw image sequences from the camera can be stored directly on the hard disk. Here, a decision about the format of the images needs to be made. Users can choose between Windows Bitmap (BMP), Tagged Image File Format (TIFF) and JPEG in the preferences of *PupilEXT*. The BMP format represents an uncompressed image format, resulting in large file size. In contrast, JPEG is a lossy compressed format commonly used in consumer photography due to its small size. The TIFF cannot be directly categorized into either of these classes, as it represents an adaptable container that can hold both compressed and uncompressed image formats. A clear-cut decision on which format to use cannot be made easily. While uncompressed formats such as BMP would result in the highest quality of images, the size of data that needs to be handled cannot be underestimated. For the use case of recording images on a disk, one needs to be able to write image data with a rate up to the camera's maximal frame rate, i.e., 120 fps.

Given the camera(s) of the reference system with a resolution of 2,048 px × 1,536 px and assuming a bit depth of 8 bits for greyscale images, the resulting image size is ≈3.15 MB. However, with 120 images per seconds, this results in a required writing speed of ≈377.49 MB/s for a single camera and ≈755 MB/s for the stereo set-up. Image size for compressed formats such as JPEG cannot be estimated this easily. Thus, an average image size observed from sample recordings of the reference system is taken. Results are greyscale images with an average size of around 840 kB. Consequently, JPEG requires a writing speed of up to ≈100 MB/s for a single camera and around 200 MB/s in a stereo camera setting. Solely based on the required writing speed without incorporating delays from, i.e., the computational overhead of compression, the speed of traditional hard disk drives (HDDs) is only sufficient for writing JPEG images in a single-camera set-up. More modern hardware in form of SATA 3, solid-state drives (SSDs) can further handle single and stereo camera set-ups for JPEG images, or just a single camera using BMP images. For recent NVMe-based SSDs, the writing speed is theoretically sufficient for writing BMP images in a stereo camera set-up. Note that the discussed rates all referred to the maximal frame rate of 120 fps. Saving images for later analysis is generally recommended for short recordings where the accuracy of the various pupil detection algorithms is of interest.

Pupil Diameter Recording

Pupil data are recorded in CSV files that store all acquired values of a pupil measurement. Pupil values can be recorded in an online measurement with connected cameras or in an offline measurement in which images are loaded in *PupilEXT* for post-acquisition evaluation. For online measurements, each pupil measurement is associated with a timestamp provided by the system time in milliseconds since Unix epoch, which is synchronized with the camera's internal hardware clock. In offline measurements, where images are read from files, no timestamp is available. Thus, the corresponding filename is used to associate each measurement. The fitted ellipse can be reconstructed from the stored ellipse parameters: width, height, center position and angle. Further recorded data for analysis are the pupil diameter, circumference and confidence measure. The pupil diameter is stated in pixel by default, and when in stereo mode, it is additionally stated in absolute units.

Regarding the pupil detection confidence, a value is only available when the applied pupil detection algorithm provides such a measure. However, a second confidence value called outline confidence is provided independently of the used algorithm. This confidence measure is based on the outline

contrast of the inner and outer regions of the fitted ellipse (Santini et al., 2018a). The goal of such value is to describe the reliability of the detected pupil diameter. These measures are useful to directly filter pupil detections that may constitute a false detection or include high uncertainty. Filtering out such detections is a common practice in the pre-processing of pupil detection results (Kret and Sjak-Shie, 2019). Santini et al. (2018a,b) apply a combination of different metrics for their confidence measure. Besides the outline confidence, the ellipse axis ratio and an angular edge spread metric are used. The ellipse axis ratio describes the ratio between major and minor axes of the ellipse, aiming to state the degree of distortion of pupil fit. The angular edge spread measures the spread of the found points on the fitted ellipse. If the points are evenly distributed, it is more likely that they originate from an exact pupil contour. We simplified the accessibility of the data by using a tabular text-based format, i.e., in the form of a CSV file. This format is independent on the used system and is commonly used for measurement recordings.

Camera Calibration

The goal of the camera calibration is to remove distortions caused by the camera lens and to estimate a projective transformation for mapping world coordinates to image coordinates. A camera projection matrix in the form of $M = K[R \cdot T]$ is used for mapping. K denotes the intrinsic parameter and $R \cdot T$ the extrinsic parameter matrices. The intrinsic matrix K projects points in the camera coordinate system to the image coordinate system with the values of the focal lengths (f_x, f_y) and the optical center (c_x, c_y) of a camera. These parameters are independent on the viewed scene and are reusable. The extrinsic matrix $[R \cdot T]$ represents the projection of world coordinates to camera coordinates, consisting of a 3×3 rotation matrix R and the 3×1 translation vector T (OpenCV, 2020). By using the camera projection matrix M, an image coordinate P_c can be projected into the associated world coordinates P_W. Such projection is typically applied in stereo vision, where the camera matrices of two or more cameras are used to estimate the depth and position of a point in world coordinates captured by these cameras. A further application of camera calibration is the correction of lens-induced distortion. Here, two types of distortion exist, radial and tangential distortions. For correcting distortions in a pinhole camera model, the calibration process estimates coefficients representing the distortions in the image, resulting in the five distortion coefficients $C = (k_1, k_2, p_1, p_2, k_3)$.

Implementing Single-Camera Calibration

In *PupilEXT*, we perform the single-camera calibration, e.g., the estimation of the camera parameters K with the computer vision library *OpenCV* library and its calibration routines. For this, a total of 30 images are collected with a rate of 0.5 fps, independently from the adjusted camera frame rate. After one image is collected, the depicted calibration pattern is detected, and feature points of the pattern were extracted. Successfully detected feature points and their positions are then stored and visualized in the calibration interface of *PupilEXT*. If the detection was not successful, the image is discarded, and the process will be applied again to the next camera image. This procedure is repeated until the specified number of images is collected. The camera calibration process is performed when enough feature points are collected. This function optimizes the camera parameters by minimizing the reprojection error according to the algorithm of Zhan (Zhang, 2000). The reprojection error describes the root mean square error (RMSE) distance between the reprojection of the observed feature points using the current camera parameters and their known position in the calibration pattern.

After successful camera calibration, the quality of the resulting calibration is an essential metric. Its quality is primarily dependent on the accuracy of the detected feature points, which is an edge detection task similar to pupil detection. We report in the *PupilEXT* interface the final reprojection error in the form of the RMSE. However, as this error constitutes a mean squared distance, it may be less intuitive for the user. Therefore, we compute an additional error using the mean absolute error (MAE), measuring the arithmetic mean of the absolute distances between the observed feature points and their projected estimation. The reprojection procedure of the MAE distance is identical to the reprojection error returned by the calibration routine. A set of ideal feature points of the calibration pattern in world coordinates are projected into the image plane using the estimated intrinsic and extrinsic parameters K, R and T. After the projection of the ideal feature point positions into their estimated image coordinates, they can be compared with the actual detected feature points in the captured image. The deviation is stated in the form of the Euclidian distance between the detected and idealized point positions, describing how well the camera parameter approximates the actual camera projection.

Validate Single-Camera Calibration

The reported reprojection error is based on the camera's projection matrix, optimized for the collected set of images during calibration. Therefore, the reprojection error may contain a bias due to overfitting. For quantifying potential overfitting, an additional verification feature is implemented in *PupilEXT*, performing the same procedure as in the calibration step but using fixed camera parameters. For this, we capture new calibration pattern images during the verification and calculate the reprojection error again, representing an unbiased approximation of the calibration quality. For instance, our prototyped single-camera system (**Figure 3D**) achieved an RMSE reprojection error of 0.341 px, where values under one pixel are commonly referred to as good calibration quality. For the MAE reprojection error, we achieved a value of 0.041 px, meaning that the average feature point coordinate was projected into the image plane with such a distance error. The verification with a new set of images showed a MAE reprojection error of 0.040 px.

In *PupilEXT*, the calibration parameters are stored to support the reuse at a later point. For this, a configuration file is saved after a successful calibration is completed. The file contains all essential information to reproduce and assess the state of the camera calibration, such as the attributes of the calibration pattern, the estimated camera parameter matrices and all projection error measures. The functionality of saving and restoring the

calibration configuration enables an additional use case, the correction of image distortions in offline pupil measurements.

Stereo Camera Calibration

Stereo vision offers the possibility of tracking the depth information and absolute pupil size from two or more images captured by cameras of known position. By using the calibration matrices M_i of two cameras, it is possible to triangulate image coordinates in both images to their corresponding world coordinates P_W. For this, matched points in both images must be found. Therefore, the pupil detection must be applied to images from both cameras (**Figure 6A**). For absolute pupil size calculation, the ends of the major axis of the ellipse are extracted and triangulated into world coordinates, and their distance was computed through the Euclidian distance (**Figure 6A**).

Triangulation determines the world position of an image point through its perspective projection in two or more images. Each projection point in an image corresponds to a projection line in world coordinates, representing all possible world coordinate positions that could have projected this point into the image. The projection lines of corresponding points can be used to determine their intersection in world coordinates. **Figure 6B** shows two corresponding image points of the main and secondary cameras (d_{p11}, d_{p21}) and their intersection point d_{pW1} in the world coordinate system. There are two challenges with this approach. First, the corresponding pupil detections in both images are required to retrieve matching points.

Second, extraction of feature points from a pupil contour may be ambiguous due to blurriness of the edge. If an identical pupil detection in both images cannot be guaranteed, potential deviations can be prevented by detecting and filtering those situations from the data stream. In *PupilEXT*, we use the corners of the minimal encompassing rectangle of the fitted ellipse (d_{p11}, d_{p21}) and (d_{p21}, d_{p22}) as feature points for triangulation (**Figure 6A**). The corner points correspond to the major axis of the ellipse for having a more robust feature selection in both images.

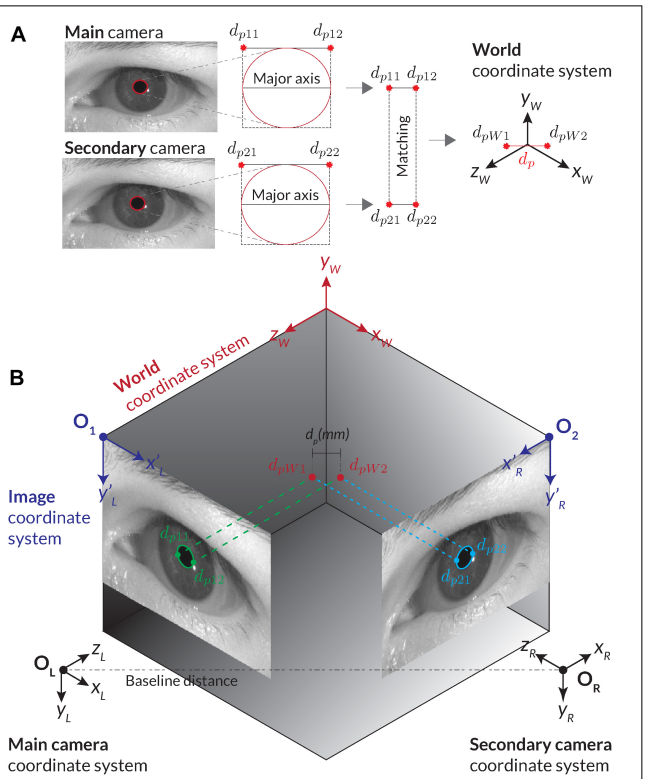


FIGURE 6 | Illustration of calculating the absolute pupil diameter with the stereo vision set-up. **(A)** For the corresponding stereo images, two pupil detections are carried out. We use the ellipses of the pupil detections and their minimal encompassing rectangle as feature points for matching. Through triangulation, the corresponding stereo images are transformed into world coordinates. The absolute pupil diameter is calculated with the Euclidian distance between the two world coordinates. **(B)** Procedure of the stereo transformation with the main and second cameras' images.

Implementation of Stereo Vision

Given the two recognized pupil ellipse results from the main and second cameras (**Figure 6B**), we check the success of pupil detection and confidence in both images. Naturally, if one of the detections failed, no matching points (**Figure 6A**) can be extracted or triangulated into the world coordinate system. In valid cases, the feature points (d_{p11}, d_{p12}) and (d_{p21}, d_{p22}) of both ellipse fits are extracted. Here, the bounding rectangle of the ellipse fit is leveraged, and the corner points from the major axis are extracted (**Figure 6A**).

Assuming the calibration parameters of both cameras are available, the paired ellipse image point coordinates (d_{p11}, d_{p12}) and (d_{p21}, d_{p22}) are corrected for potential distortions using the distortion coefficient matrices. Next, the corresponding image feature points (d_{p11}, d_{p21}) and (d_{p12}, d_{p22}) are triangulated using the OpenCV function *cv::triangulatePoints*. The triangulation results P_{H1} and P_{H2} are represented in homogeneous coordinates, which then are converted into Cartesian coordinates (Eqs. 1 and 2).

$$P_H = \begin{bmatrix} X_H \\ Y_H \\ Z_H \\ W_H \end{bmatrix}, \text{home2cart}\,(P_H) = \begin{bmatrix} X_H/\omega \\ Y_H/\omega \\ Z_H/\omega \end{bmatrix} \quad (1)$$

$$\omega = \begin{cases} W_H, & \text{if} W_H \neq 0 \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

With the transformed points in the world coordinate system (d_{pW1}, d_{pW2}), we determine the absolute pupil diameter through the Euclidian distance (**Figure 6B**). In the experiments, the computation time of this procedure (feature extraction, distortion correction and triangulation) was on average 0.03 ms, which should not significantly influence the maximum possible processing rate of pupil measurements.

However, no further criteria are applied for checking the reliability of the stereo vision result, as it is left open for the user applying the post-processing procedure. We did not consider a general threshold for pre-filtering to be necessary since the user should have full control over the evaluation of the data. For

this, we provide all necessary raw data from both cameras in the recorded CSV file.

Calibration of Stereo Vision

A requirement for the stereo triangulation is the projection matrices M_i of both cameras. As discussed in the *Camera Calibration* section, the parameters of a single camera are estimated in the calibration procedure, resulting in the intrinsic parameters of the cameras. As the projection matrix M consists of both the intrinsic and extrinsic parameters, the extrinsic parameters are estimated through a OpenCV stereo calibration procedure, which takes the intrinsic parameters of each camera, returning the extrinsic parameter in the form of the rotation matrix R and the translation matrix T. Thereby (R, T) describe the relative position and orientation of the main camera with respect to the secondary camera coordinate system (OpenCV, 2020). After the estimation of these extrinsic parameters, the projecting matrices M_1, M_2 can be calculated with the equation $M = K[R \cdot T]$. Notably, in a stereo camera set-up, the main camera is typically selected as the origin of the stereo camera coordinate system. Thus, the projection matrix of the main camera does not apply rotation or translation and is therefore given by $M_1 = K \cdot [I|0]$, where T is replaced with the identity matrix I and R is replaced with the zero vector.

Validate Quality of Calibration

Similar to the single-camera calibration, the reprojection error is returned as RMSE by the stereo calibration procedure. In stereo vision mode, the reprojection error states the distance between the observed and reprojected feature points combined for both cameras in image coordinates. However, for the user, it would be more useful to be able to assess the quality of the stereo calibration in terms of absolute units. Therefore, we leveraged the predefined size of the calibration pattern to calculate the measurement error of the calibration in absolute units. For this, we measure the absolute square size of, i.e., the chessboard pattern, using the detected feature points from both cameras in the calibration routine. The detected feature points of the calibration pattern are undistorted, stereo triangulated and converted into Cartesian world coordinates.

Next, the measured square size is compared with the known distance between two corner feature points of the calibration pattern. As a result, we report the calculated error of the stereo camera system in absolute units calculated by the distance deviation between the measured and idealized sizes of the pattern. However, the stated error again could be biased by the overfitting in the calibration routine. Therefore, we implemented a verification routine that checks the absolute measurement error using a new set of images with the calculated projection matrices. Similar to the single-camera mode, the stereo calibration matrix can be saved and loaded into the software for the next usage, reducing new calibration effort. Here, we recommend verifying the old calibration before a pupil measurement is conducted. If the lens settings or camera position are slightly changed, the transformation matrix needs to be re-created by a new calibration procedure. The necessity can be quickly checked using the verification function in *PupilEXT*.

Performance of PupilEXT

The performance of *PupilEXT* in pupil measurements depends on various factors such as processing power of the system, frame rate of the camera and the applied pupil detection algorithm. As listed in **Table 1**, the runtimes of the pupil detection algorithms vary significantly. For the goal of conducting pupil measurements with a frame rate of 120 fps, a maximal runtime of around 8 ms or less is necessary. Additional computations such as correcting lens distortion can increase the needed computation time per image. We optimize the computational complexity in *PupilEXT* by using a region of interest (ROI), reducing the amount of pixel that needs to be processed. The ROI can be adjusted interactively by the user in the interface.

In combination with the *PuRe* pupil detection algorithm, we achieved a stable pupil measurement at 120 fps on full images. With manually specified ROI selection, the frame rate can be pushed further, as *PupilEXT* is completely implemented in C++, supported by parallel computation using CPU threads.

The Graphical User Interface of PupilEXT

Figure 7 illustrates the GUI of *PupilEXT* during a pupil measurement in the stereo camera mode. Via the taskbar of the GUI (**Figure 7**, points 1 to 9, blue), the essential function of the software is linked. Before a pupil measurement, the camera mode and the respective cameras must be selected to establish a connection (**Figure 7**, point 1, blue). In the camera settings also a connection to the microcontroller can be established if a hardware trigger is required. After successful connection to the cameras, a window with a live image view of the cameras is opened. Camera parameters such as gain factor, exposure time or maximum frame rate can be changed at any time via a quick start button (**Figure 7**, point 3, blue). Next, one of the six pupil algorithms can be selected in the pupil detection preferences (**Figure 7**, point 1, green). In addition to the algorithms, the parameters of the method can be set to optimize the detection accuracy when necessary (**Figure 7**, point 3, green). We have provided a preset of parameters that can be selected (**Figure 7**, point 2, green). In addition to the standard parameters from the original papers, we have added optimized values that are adapted to different ROI sizes. We have set the *PuRe* method as a standard method in *PupilEXT*.

The pupil detection of the captured live images can be started with the eye symbol in the main window (**Figure 7**, point 4, blue). We provided in the live view window a quick action menu (**Figure 7**, point 1, red), which can be used to adjust the image size, setting the ROI or displaying magnification of the pupil. The ROI features allow placement of a rectangular area over the eye to improve performance further when recordings at a higher frame rate of 120 Hz are needed. Note that for the stereo camera mode, a calibration should be carried out; otherwise, the absolute pupil diameter will not be available. The calibration window can be reached through the taskbar in the main window (**Figure 6**, point 5, blue).

In the calibration window (**Figure 7**, point 4, green), one can select the type of calibration pattern. Next, the calibration can be started, resulting in the calibration file that is saved

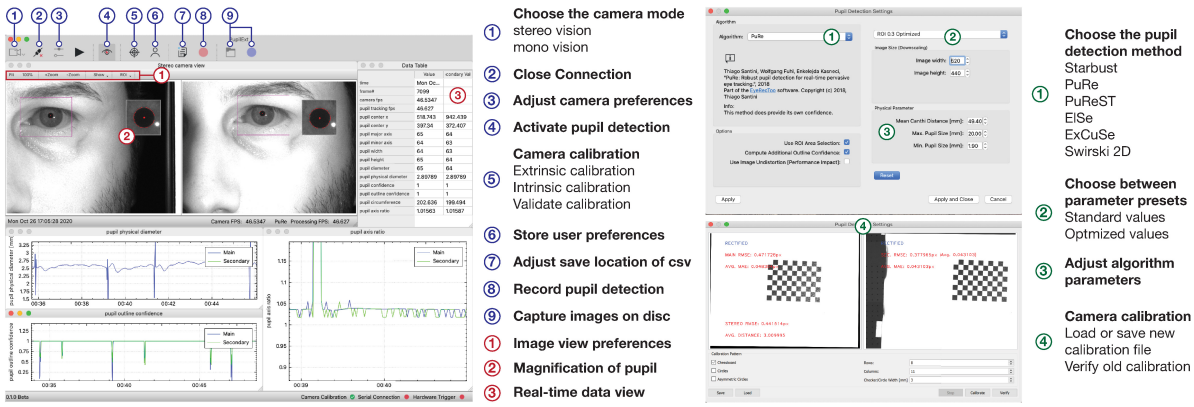


FIGURE 7 | The graphical user interface of the programmed software *PupilEXT* during a pupil recording in stereo camera mode. In the main window's taskbar, various functions can be accessed for quick actions. In the pupil detection sub-window, a pupil algorithm with respective parameters can be adjusted. The camera calibration can be done directly in *PupilEXT*. Calibration files can be saved and validated to give an outline of the camera system's edge detection accuracy, which is essential for a pupil measurement pipeline.

locally on the hard disk. If a calibration file already exists, it can be loaded via the calibration window (**Figure 7**, point 4, green), and its validity can be again verified. The stated calibration accuracy can be recorded in a CSV file during the validation procedure.

After the calibration is completed, the absolute pupil diameter is displayed in the data view, which also lists all tracked pupil values in real-time (**Figure 7**, point 3, red). Each of these values can be visualized in a real-time plot by selecting the specific value in the data view. For recording the pupil measurements, a disk location can be selected to save the pupil data in a CSV file (**Figure 7**, point 7, blue). The data can be saved continuously with the recording button (**Figure 7**, point 8, blue). The raw images can be saved with the blue recording button (**Figure 7**, item 9, blue) for later offline pupil detection in *PupilEXT*. In the **Supplementary Materials**, we have added hands-on video materials to illustrate the pipeline of usage and the features. Additionally, we offer the feature of creating and loading custom profiles (**Figure 7**, point 6, blue), which opens the software in a specified state to avoid the workload when *PupilEXT* is started next time.

DEMONSTRATION OF A MEASUREMENT PIPELINE WITH PupilEXT

To illustrate the measuring procedure with *PupilEXT*, we performed an exemplary experiment on the wavelength-dependent pupil light response. We recorded the pupil diameter of an observer with six repetitions (trials) using *PupilEXT*, while different light spectra were turned on at a steady luminance. For this, a subject looked into a 700 mm × 700 mm sized homogeneously illuminated observation chamber. The illumination was generated by a custom-made temperature-controlled (30°C ± 0.1°C) multi-channel LED luminaire, which was used to trigger the pupil diameter with chromatic stimulus spectra (Zandi et al., 2020). Pupil foreshortening error (Hayes and Petrov, 2016) was minimized by using a chin rest for positioning the subject's head. Additionally, the gaze point was fixed with a 0.8° sized fixation target (Thaler et al., 2013) in the middle of the adaptation area. On the left eye's optical axis, a stereo camera system consisting of two Basler acA2040-120um cameras with 50-mm lenses was set up (**Figure 3A**).

The pupil diameter was triggered using chromatic LED spectra with peak wavelengths λ_{Peak} of 450 nm [full width at half maximum (FWHM): 18 nm, **L** = 100.4 cd/m^2 ± SD 0.23 cd/m^2) and 630 nm (FWHM: 16 nm, **L** = 101 cd/m^2 ± SD 0.31 cd/m^2], which were switched on for 30 s. Before each stimulus spectrum, a phosphor-converted white-colored LED with a correlated color temperature of 5,500 K (**L** = 201 cd/m^2 ± SD 0.48 cd/m^2) was presented to adapt the pupil diameter to its baseline. The order of the chromatic stimulus spectra was randomized. One pupil measurement trial lasted 240 s, as the anchor spectrum (5,500 K) was switched on twice between each chromatic stimulus for 90 s, and the main stimuli (450 and 630 nm) were switched on 30 s. The spectra were measured 20 times before and after the experiment using a Konica Minolta CS2000 spectroradiometer. We controlled the luminaire with a custom-made MATLAB script, which stored the switch-on times of the spectra in a CSV-File. Possible switch-on latency times during the command transmission from MATLAB to the luminaire's hardware were taken into account by tracking the processing time in the embedded software. We recorded stereo eye images with 30 fps (*.bmp) during each pupil examination trial (240 s), making it possible to detect the pupil diameter from the images with different detection algorithms, later on using the offline pupil analysis mode of *PupilEXT*. The pupil data were synchronized with the luminaire's switch-on times afterward using a MATLAB script.

Pre-processing the Measured Raw Data

Recorded raw pupil data are usually occupied by artifacts or other non-physiological pupil changes that need to be pre-processed

(**Figure 8A**). For the pupil data recorded by *PupilEXT*, we recommend a two-step filtering procedure. First, every data point that has an outline confidence measure (Santini et al., 2018a) lower than 1 should be left. With this step, artifacts caused by eye blinks are detected robustly (**Figure 8A**). Other artifacts can occur if the matching points (**Figure 6B**) between the first and second cameras differ, resulting in a non-physiological shift of the pupil diameter, visible through slight peaks in the data. We identify matching point errors by comparing the stated axis ratio of the ellipses between the main and second cameras. The axis ratios differ because of the second camera's positioning causing a perspective pupil area change. However, the ellipse axis ratio difference between the ellipses of cameras 1 and 2 should remain constant within a certain range. Thus, the reliability of the matching points (**Figure 6B**) can be detected by calculating the difference of the axis ratio across the data points and removing all strong outliers from the sample dataset (**Figure 8A**). We have pre-processed the recorded pupil data according to this two-step procedure. The results of one raw pupil measurement trial (240 s) using the *PuRe* algorithm and respective pre-processed pupil data are shown in **Figure 8B**. Eye blinks can approximately be tracked by identifying the outline confidence areas that fall below one. However, an eye-blink detection via the outline confidence measure can only work if the algorithm's detection rate is robust; i.e., the pupil is detected in more than 90% of valid eye image cases. We implemented the proposed two-step pre-processing method in MATLAB. The script is available on the GitHub repository of the *PupilEXT* project. Additionally, the recorded eye images are made available online together with the stereo calibration file. The data can directly be loaded into *PupilEXT* for a hands-on experience.

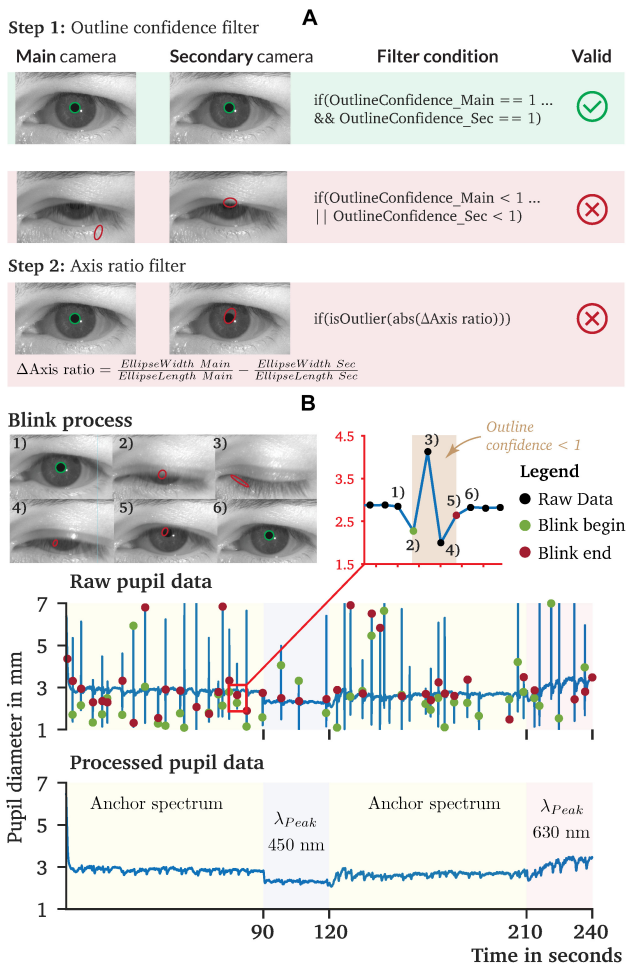


FIGURE 8 | Recorded pupil data and proposed pre-processing procedure of pupil diameter data collected by *PupilEXT*. **(A)** A two-step pre-processing procedure is proposed, which uses the outline confidence and the axis ratio of the cameras' tracked ellipses. **(B)** Recorded pupil data from our sample experiment to illustrate the performance of *PupilEXT*. The outline confidence can be used to identify eye occlusions in the data approximately. The two-step pre-processing can remove artifacts and other unnatural physiological pupil diameter changes.

Comparison of the Pupil Detection Approaches

A majority of pupil detection algorithms was evaluated based on their accuracy in estimating the pupil center (**Table 1**), as they are mainly intended for eye-tracking applications. One of the works evaluating the pupil fit was Świrski et al. (2012) in which their approach was compared against the Starburst algorithm. The pupil fit was assessed utilizing hand-labeled pupil measurements and the Hausdorff distance. The Hausdorff distance (Rote, 1991) thereby describes the maximum Euclidean distance of one ellipse to any point on the other ellipse (Świrski et al., 2012). Results show that the *Swirski* algorithm improves the detection rate for a five-pixel error threshold from 15% for Starburst to 87%, showing that not every eye-tracking algorithm is suited for pupil measurements. Fuhl et al. (2015) evaluated the *ExCuSe* algorithm, comparing their approach with the *Swirski* and *Starburst* algorithms. However, only the distance between the pupil center estimation and ground-truth was evaluated. The evaluation was performed on 18 datasets of pupil images captured under highly challenging real-world conditions. The detection rate for a five-pixel error threshold shows an average rate of 17% for *Starburst*, 40% for *Swirski* and 63% for *ExCuSe*.

A similar evaluation was repeated in the works of *ElSe* (Fuhl et al., 2016a), *PuRe* (Santini et al., 2018a), and *PuReST* (Santini et al., 2018b), where they conducted evaluations using overlapping datasets and the pupil center distance as a performance value. Within a five-pixel error threshold, the algorithm of Starburst shows a detection rate of 13.44, 28 to 36% for *Swirski*, 50 to 58% for *ExCuSe*, 66 to 69% for *ElSe*, 72% for *PuRe* and 87% for *PuReST*. In these evaluations, a performance loss for highly challenging recorded images was observed. Specifically, images with low-intensity contrast and pupils containing small reflections impaired the pupil detection algorithms. Santini et al. showed that the average runtime of the *PuReST* algorithm is 1.88 ms, compared with *PuRe* with 5.17 ms (Santini et al., 2018b), making *PuReST* the fastest approach with the highest pupil center detection rate. Note that these results apply to images that do not occur under laboratory conditions. Topal et al. (2017) evaluated the *APPD* algorithm (Topal et al.,

2017) together with *Starburst*, *ElSe* and *Swirski*. The pupil fit and processing time were used to quantify the performance of the algorithms. For the pupil fit, the pupil localization was used, which quantifies the overlap ratio between the detected ellipse and the ground-truth, stated as [0, 1]. The results indicate a high pupil localization of 0.97 for *APPD* compared with 0.93 for *Swirski*, 0.92 for *ElSe* and 0.77 for Starburst. Additionally, Topal et al. measured an average computation time of 5.37 ms for *APPD*, 7.12 ms for *Else* (7 ms), 47.17 ms for *Swirski* (3.77 ms) and 49.22 ms for *Starburst* (100 ms). The numbers in parentheses define the originally reported runtime of the respective algorithms.

Based on the literature, it can be stated that *PuReST* is the top performer when evaluating the pupil's center detection rate with highly noisy images. However, these results represent the detection rate with a five-pixel error threshold and do not state the accuracy of their pupil size measurements. Only the evaluation of Topal et al. (2017), Świrski et al. (2012) carried out a performance test on the pupil fit. Their results state a different picture, with *Swirski* performing better than *ElSe*.

Another aspect that could significantly affect the performance of a pupil detection algorithm is the parameters' count. Each algorithm has a set of parameters that need to be tuned by the user to match the image composition. Selecting appropriate values may constitute a challenge for the user. Thus, the fewer parameters an algorithm possesses, the simpler its application. Comparing the number of parameters of the pupil detection algorithms, *Swirski* includes 11, followed by *Starburst* with five and *PuRe* and *PuReST* with three. *Else* and *ExCuSe* have only two parameters. We have stored in our proposed software *PupilEXT* the standard values of the algorithms as stated by the authors and additionally optimized three sets of parameters for pupil measurement applications under different image compositions.

Validation of the Pupil Detection Algorithms

We evaluated the captured eye images from our pupil experiment using the six available pupil detection algorithms in *PupilEXT*. Ideally, the pupil diameter should remain steady across the detection algorithms, as the same eye image sets were used for evaluation. However, due to the algorithms' different parameters settings and approaches, the measured diameter may differ. In **Figure 9A**, we have plotted the detected raw pupil diameter from one experimental trial (240 s) to illustrate how differently the algorithms perform based on the same acquired image set. For each raw data plot panel, the respective pre-processed pupil data are illustrated, which were obtained using the proposed two-step method. The *ElSe*, *ExCuSe*, *PuRe* and *PuReST* algorithms achieved an acceptable pupil detection rate, visually noticeable through the lower number of artifacts in the respective raw dataset (**Figure 9A**). As discussed, the artifacts in the raw data can be filtered by removing the detected pupil diameter with an outline confidence of less than 1. In **Figure 9B**, we illustrated a sample of recorded pupil images with the respective outline confidence, showing that an invalid pupil fit can be detected and removed when using such a metric.

The *Starburst* algorithm caused a higher number of artifacts. Subsequent pre-processing of the raw data using the two-step method was not helpful, as the *Starburst* algorithm caused too many false detections. The *Swirski* algorithm had difficulties in detecting small pupil diameter at the 450-nm stimulus. After the invalid pupil data were filtered from the 450-nm time frame, there were almost no valid data left for linearly interpolating the missing values. Also, the *Swirski* algorithm had no robust detection rate for the pupil recording with the 630-nm spectrum. However, the cameras' lenses were equipped with optical IR-high-pass filters so that the spectral-dependent detection quality was not due to the type of light spectrum. Each pupil detection algorithm has a certain number of parameters that need to be adjusted depending on the image resolution or how large the pupil is in relation to the image size. An incorrect combination of parameters could affect the pupil detection at differently sized diameters, as the algorithm itself could rule out smaller pupils.

The proposed technique for detecting eye blinks based on an outline confidence (**Figure 8B**) is highly affected by the detection rate. For example, it is no longer possible to distinguish between a false pupil fit or a closed eyelid at a higher rate of pupil detection artifacts (**Figure 9A**). Additionally, the *ExCuSe* algorithm offers a threshold value that can be used to detect eye blinks. In this way, values that indicate a closed eyelid will automatically be removed by the respective pupil detection algorithm itself, leading to the fact that a subsequent analysis of eye blinks is no longer possible. Therefore, an eye-blink recognition using the outline confidence seems to work well only with *PuRe* and *PuReST*.

In **Figure 10**, we calculated the average percentage of the invalid data rate for each algorithm and spectrum separately to illustrate the pupil detection algorithms' performance across the conducted pupil measurement trials. The invalid data rate is defined as the number of diameter values that had to be removed from the raw dataset when using the two-step pre-processing approach (**Figure 8B**). The *ElSe*, *ExCuSe*, *PuRe* and *PuReST* algorithms had a lower invalid data rate of 10%, indicating good detection performance across all measurement trials (**Figure 10**). The *Starburst* algorithm failed to perform a valid pupil fit at 450 nm in 58.46% SD 5.93% of cases. At the second reference spectrum (5,500 K), the pupil detections from *Starburst* failed in 36.82% SD 7.1% of the cases. Since the invalid pupil detection rate was higher than 10% for every stimulus spectrum, we assume that the performance of *Starburst* is independent of the parameter settings; possibly, the false pupil fits arise due to the contrast or resolution in the eye image. The *Swirski* algorithm's performance suffered mainly at the 450-nm stimulus with an average error rate of 81.09% SD 10.10%. This behavior seems to be due to the algorithm's parameters adjustments, as the invalid data rate is higher for smaller pupil diameters. Our results are in line with previous benchmarks from the literature, which showed that the *Starburst* and *Swirski* algorithms had lower detection rates (Fuhl et al., 2015, 2016a; Santini et al., 2018a,b). Note that the *Swirski* algorithm could have a better pupil fit, as it does not downscale the eye images before processing (Świrski et al., 2012; Topal et al., 2017). Since the *Swirski* algorithm has 11 free parameters that need to be adjusted, it is not a practical algorithm in our view because the detection method could suffer its robustness

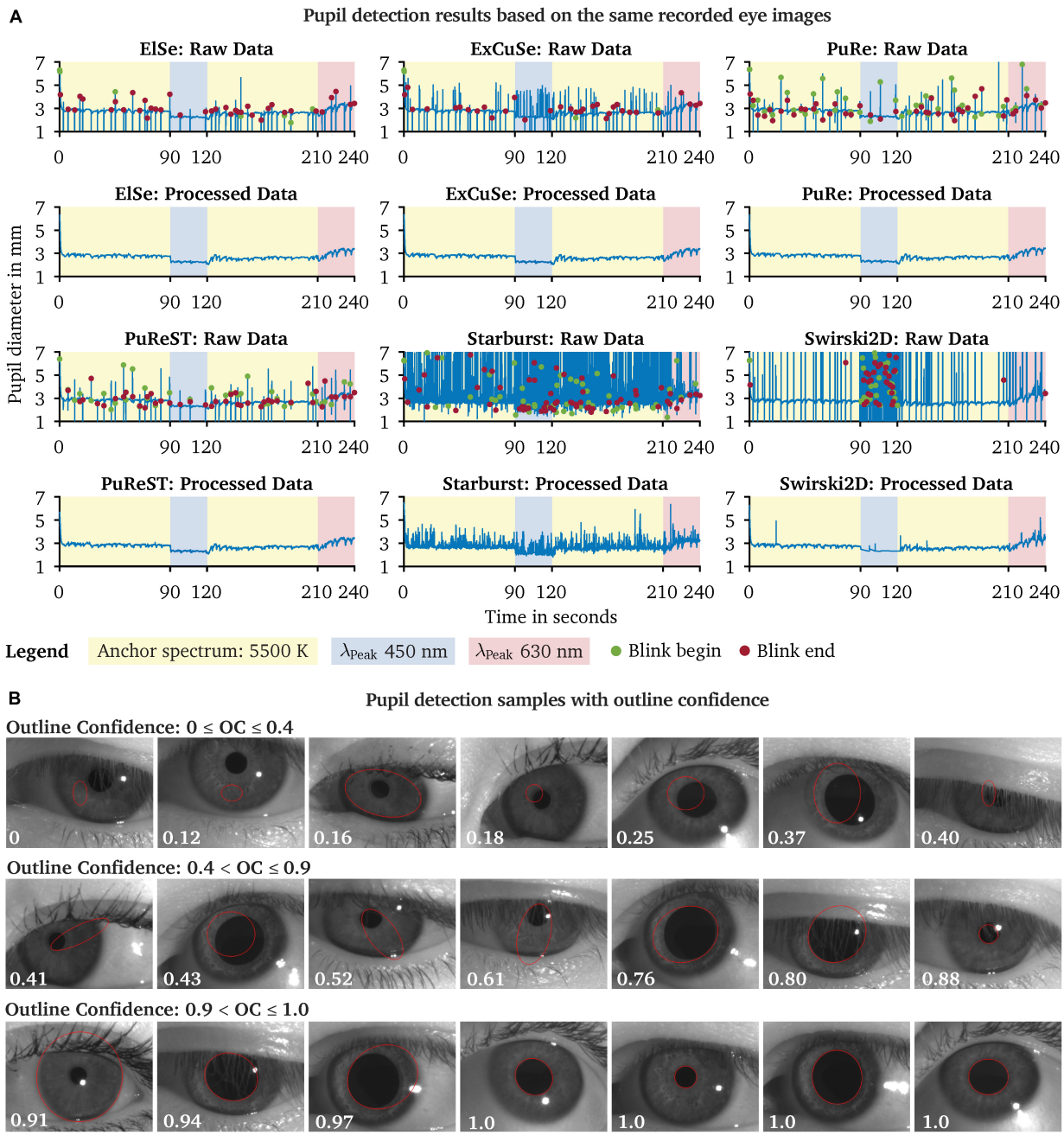


FIGURE 9 | Comparison of the pupil detection algorithms based on the same eye image set and visualization of the pupil ellipse fit as a function of the outline confidence. **(A)** Eye images from one subject were recorded during a chromatic pupillometry experiment using *PupilEXT*. The pupil was exposed to LED spectra of the peak wavelengths 450 nm (L = 100.4 cd/m^2 ± SD 0.23) and 630 nm (L = 101 cd/m^2 ± SD 0.31) for 30 s. An anchor spectrum with a correlated color temperature (CCT) of 5,500 K (L = 201 cd/m^2 ± SD 0.48) was turned on for 90 s between each stimulus. The pupil diameter from the recorded images was extracted using the available algorithms in *PupilEXT* and pre-processed to illustrate the algorithms' detection differences. **(B)** For each detected diameter, an outline confidence measure is provided and used as an indicator to filter unreliable pupil fits from the dataset. Pupil fits from different measurement sessions are illustrated as a function of the outline confidence. We recommend discarding all pupil diameters with a lower outline confidence measure of 1.

when using the wrong settings. The advantage of the pupil algorithms *ElSe*, *ExCuSe*, *PuRe* and *PuReST* is the smaller number of parameters that need to be set, leading to less error-proneness and practicability in conducting pupil measurements.

To better estimate how much the pupil diameter deviates depending on the used pupil algorithm, we evaluated the acquired eye images with the top-performing algorithms (*ElSe*, *ExCuSe*, *PuRe* and *PuReST*) and calculated the steady-state

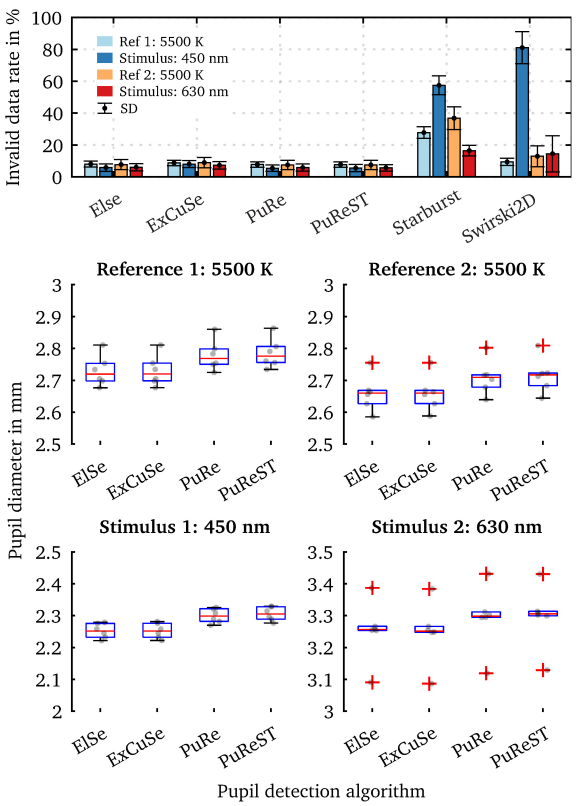


FIGURE 10 | Percentage of invalid pupil fits inside the raw data, and the averaged steady-state pupil diameter of the last 5 s. Pupil data are from one subject with six repetitions in each condition. **(A)** Mean of the invalid data point count (outline confidence < 1) in per cent for each algorithm and used spectrum. A higher invalid data rate indicates that more raw data need to be removed due to inaccurate pupil fits. The *Starburst* algorithm did not provide an adequate detection rate, mainly observed at 450 nm (57.46% ± SD 5.93%) and the second reference spectrum (36.82% ± SD 7.1%). The *Swirski* algorithm showed a significant invalid detection rate of 81.09% ± SD 10.10% at 450 nm. Since the cameras were equipped with an IR-high-pass filter, the spectral-dependent pupil detection rate is mainly due to the differently sized pupil diameter. Due to the increased number of parameters from *Starburst* and *Swirski*, a generalization for small and large pupil diameters is more challenging. **(B)** The temporal pupil diameter was averaged over the last 5 s to compare how differently the pupil algorithms evaluate the same dataset. The *ElSe* and *ExCuSe* algorithms have approximately the same pupil diameter in all scenarios, which is due to the computation method's similarities. The same applies to the *PuRe* and *PuReST* algorithms. The measured pupil diameter's uncertainty range is on average 0.05 mm ± SD 0.004 mm, originating from the different detection results with the same dataset.

equilibrium pupil diameter. For this, we calculated the pupil diameter's mean value over the last 5 s of a measurement. **Figure 10B** shows the steady-state pupil diameters from the six measurement trials. The scatter within a pupil algorithm is due to the pupil diameter's intrasubject variability, which is mainly induced by cognitive effects and can be up to 0.5 mm (Zandi et al., 2020; Zandi and Khanh, 2021). The absolute mean pupil diameter differences between the *ElSe* and *ExCuSe* algorithms are negligible with $6 \cdot 10^{-4}$ mm at 450 nm and 0.0041 mm at 630 nm, which are due to the same detection approaches. The same was applied for the *PuRe* and *PuReST* algorithms with an absolute mean diameter difference of 0.0061 mm at 450 nm and 0.0051 mm at 630 nm. The *PuReST* algorithm was an extension of *PuRe*, allowing faster pupil detections and explaining the similar pupil fits. However, the mean difference between the algorithm groups *ElSe*/*ExCuse* and *PuRe*/*PuReST* is 0.054 mm SD 0.0043 mm. This is particularly interesting because it indicates how much the measured pupil diameter can deviate when different detection method approaches are applied to the same eye image set. Therefore, in cognitive studies in which the pupil diameters' mean difference is less than 0.1 mm, we highly recommend reporting the algorithm and respective parameter settings. The parameters that we used for our pupil detection experiments are stored in the *PupilEXT* software and also available on the GitHub repository of this project.

Determining the Pupil Measurement Accuracy

The accuracy of the pupil measurement can by characterized with *PupilEXT* by two approaches. First, the validation process of the stereo calibration determines the quality of the system, indicated by the reprojection error in MAE within *PupilEXT*. However, such a metric does not include the inaccuracies caused by pupil detection methods. Therefore, it is advisable for checking the validity of the system by a circular formed reference object. For this, we placed a reference object of known size (5 mm) in front of the subject's eye and determined the diameter using a pupil detection algorithm in *PupilEXT* (**Figure 11**).

The measured raw data of the reference object showed a MAE of 0.014 mm. After pre-processing the data with the two-step method, a MAE accuracy of 0.0059 mm was achieved

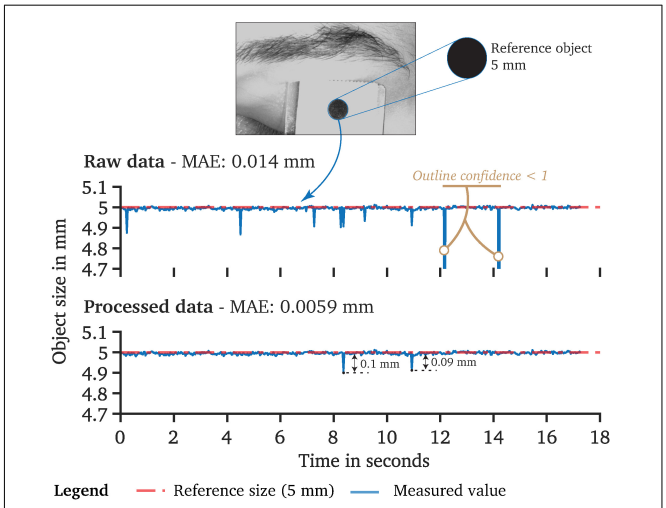


FIGURE 11 | Determination of the used stereo camera system's measurement accuracy. A reference object of known size (5 mm) was placed in front of the observers' occluded eye. The diameter of the object was tracked using the *PuRe* algorithm. Based on the raw data, a mean absolute error (MAE) of 0.014 mm can be determined. However, the raw data contain artifacts that we have removed using the proposed two-step pre-processing approach. After pre-processing, we can state a measurement accuracy of 0.0059 mm (MAE).

with our prototyped system. It should be noted that such a measurement accuracy is still an idealized approximation since the reference object was kept still without interference. After pre-processing, isolated peaks remained with an amplitude of 0.1 mm. However, remaining pupil data are usually smoothed, making such remaining isolated peaks negligible.

Limitations of the Proposed Pupillometry Toolbox

The current version of *PupilEXT* offers a comprehensive solution for pupillometry. However, the software is not designed for two-eye measurements, as only one eye at the same time can be captured. We recommend positioning the ROI in the live view of *PupilEXT* software over one eye to let the algorithms iterate inside the specified region if two eyes are visible in the image. Furthermore, an online pupil measurement can only be carried out with Basler branded cameras. In the future, the integration of other camera brands is possible through the implemented camera class. However, externally acquired images from other camera brands can be loaded into *PupilEXT* for offline pupil detections, making it possible to use the software even without purchasing a Basler camera.

Currently, the implemented pupil algorithms perform their computations on the CPU. Therefore, we recommend using the *PuRe* or *PuReST* algorithm for real-time pupil measurements with a higher frame rate between 60 and 120 fps, as the detection approaches shine with low processing times. In the future, it would be desirable to perform calculations directly on a graphics processing unit (GPU) during an online measurement, making higher frame rates for all integrated pupil detection methods possible. Note that we did not implement a limiting threshold of the frame rate level inside the *PupilEXT* software. The frame rate is limited by the respective pupil detection algorithm's processing time, which can vary depending on the used computer. If the frame rate is too high for the computer during an online pupil measurement, the images will be stored in the machine's memory buffer and fed to the pupil algorithm one by one. In such cases, there is the risk that the working memory will overflow when operating *PupilEXT* for longer times in such a mode. Therefore, the camera fps should ideally be on the same level as the processing fps. Both metrics are stated in the live view panel of *PupilEXT*. Note that on our computer (Intel Core i7-9700K), we performed pupil measurements in stereo mode at 120 Hz without any issues when using the *PuReST* or *PuRe* algorithm. Even higher frame rates are possible in the single-camera mode because only the image from one camera has to be processed. Alternatively, eye images can be captured on the disk for later pupil detection, allowing higher frame rates. This function is available for both mono and stereo camera modes.

DISCUSSION

The idea of replacing commercial systems with open-source solutions is currently pushed by working groups topically working on eye-tracking devices (Santini et al., 2017; Arvin et al., 2020). The advantage of eye-tracking research is that standardized metrics exist that reflect the accuracy of a detected gaze point (Holmqvist et al., 2012). In pupillometry research, metrics on the pupil fit's measurement accuracy is usually not stated, mainly because most applied systems do not allow manual verification after conducted experiments. The lack of missing pupil fit metrics in commercial eye-tracking systems applied for pupil measurement motivated recent works, attempting to develop procedures or provide at least pupil measurement error information of widely used systems (Klingner, 2010; Gagl et al., 2011; Brisson et al., 2013; Hayes and Petrov, 2016; Murray et al., 2017; Wang et al., 2017; Titz et al., 2018; Coyne et al., 2019). Mathematically, the pupil center's accuracy detection is just an indicator for a good pupil fit but does not ensure it. For example, the pupil center can be correct for cases in which the gaze point differs from the camera's optical axis (eye rotation), but the detected pupil diameter can be estimated incorrectly due to the perspective distortion of the pupil image (pupil foreshortening error) (Hayes and Petrov, 2016). Additionally, it is not directly possible to reproduce the pupil fit's accuracy from the pupil center accuracy, which is mainly stated in the datasheet of eye-tracking devices. Suppose studies indicate an effect on the pupil diameter of 0.5 mm. In that case, ideally, there should be a procedure to verify that both the camera system and the applied pupil detection method can detect such small diameter margins. For example, the recently published work "Standards in Pupillography" (Kelbsch et al., 2019) rarely paid attention to possible technical- and software-induced measurement errors, although this could highly affect the validity and conclusions of research results. By comparing the pupil detection algorithms, we showed that a measurement error of up to 0.05 mm could occur with identical eye images, induced solely by the type of used detection algorithm itself. In commercial systems where it is usually unknown which pupil detection algorithm is applied, comparisons between study results in such a measurement range are difficult. Therefore, the camera's spatial resolution specification or the pupil center's measurement accuracy is insufficient for pupil measurements. From our perspective, a uniform measurement platform is essential for pupillometry, ensuring comparability and reproducibility. By verifying our proposed *PupilEXT* set up with a reference object, we offer the possibility to test and state the accuracy of the pupil's fit directly. Furthermore, the proposed system ensures reproducing pupil examinations results by using the captured images in the offline analysis mode of *PupilEXT*.

With *PupilEXT*, we have developed the first freely accessible integrated end-to-end pupil measurement platform consisting of hardware and software for professional pupillometry research in vision science. Pupil measurement can be carried out in a one- or two-camera mode. The calibration and validation procedure in *PupilEXT* are intended to provide a transparent way in reporting the measurement accuracy of a conducted pupil study. The specification of measurement accuracies is currently a major issue in pupil research since only in few publications is the validity of the pupil tracking's accuracy stated. This is mainly due to the use of commercial systems that usually do not

support validation procedures of pupil measurement pipelines. The complete software, embedded code and printed circuit board (PCB) layout of the NIR illumination are provided as an open-source project. We provide three **Supplementary Videos** to illustrate the handling of *PupilEXT*. The instruction, details about the installation and video tutorials can be found at the project's website (see text footnote 1).

As a next step, it is planned to add a gaze calibration routine to *PupilEXT* to support eye-tracking applications. Currently, we only support Basler branded cameras, but it is possible to add additional industrial camera types into *PupilEXT* since the camera access is separated from the core of the proposed software. The feature of determining the pupil diameter from externally captured images could perhaps make *PupilEXT* a standardized measurement suitable for pupil research. For this aim, we will investigate in the next studies the tracking accuracy of the integrated pupil algorithms with ground-truth images, leading to a better estimation of real-world inaccuracies under laboratory conditions.

DATA AVAILABILITY STATEMENT

The software PupilEXT as well the embedded program of the microcontroller, and PCB layout of the NIR illumination are available on the following GitHub page: https://github.com/openPupil/Open-PupilEXT.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Technical University of Darmstadt. The patients/participants provided their written informed consent to participate in this study. Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

BZ had the initial idea, supervised the project, and designed the first concept of the pupillometry platform. ML was the core software developer with contributions and supervision of BZ. BZ wrote the original draft of the manuscript with contributions of ML. ML provided the literature research and summary of existing pupil detection algorithms under the supervision of BZ. BZ created the figures with contributions of ML and AH. BZ developed and programmed the hardware trigger concept. GS, AH, TK, ML, and BZ performed review and editing of the original draft. GS, AH, TK, ML, and BZ developed the testing methodology. ML and BZ performed testing. All authors read and agreed to the submitted version of the manuscript.

FUNDING

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 450636577.

ACKNOWLEDGMENTS

We thank the German Research Foundation (DFG) by funding the research (Grant Number 450636577). This project was made possible by the outstanding previously published open-source projects in the field of pupil detection and eye tracking. Therefore, we would like to thank the authors of the ground-breaking algorithms *PuRe*, *PuReST*, *ElSe*, *ExCuSe*, *Starburst* and *Swirski*, who made their methods available to the public. We have to thank Wolfgang Fuhl, Thiago Santini, Thomas Kübler, EK, Katrin Sippel, Wolfgang Rosenstiel, Li, D. Winfield, D. Parkhurst, Lech Swirski, Andreas Bulling, and Neil Dodgson for their open-source contributions, which are part of *PupilEXT*. Additionally, we would like to thank the outstanding developers of the software *EyeRecToo*, whose open-source eye-tracking software inspired us for this work. We used the implementation of the *EyeRecToo*'s pupil class and the integrated detection methods for *PupilEXT*. We appreciate the contributions of Paul Myland, who supported us as a co-supervisor in a bachelor thesis, which topically worked on one part of this project. We highly welcome the contribution of Mohammad Zidan for the mechanical construction of the stereo camera system and the NIR illumination, which was done during his bachelor thesis, supervised by BZ. Finally, we would like to thank Felix Wirth and Thomas Lautenschläger, who joined us as student assistants in the initial phase of the project.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2021.676220/full#supplementary-material

REFERENCES

Allen, A. E., Martial, F. P., and Lucas, R. J. (2019). Form vision from melanopsin in humans. *Nat. Commun.* 10, 1–10. doi: 10.1038/s41467-019-10113-3

Aminihajibashi, S., Hagen, T., Andreassen, O. A., Laeng, B., and Espeseth, T. (2020). The effects of cognitive abilities and task demands on tonic and phasic pupil sizes. *Biol. Psychol.* 156:107945. doi: 10.1016/j.biopsycho.2020.107945

Arvin, S., Rasmussen, R., and Yonehara, K. (2020). EyeLoop: an open-source, high-speed eye-tracker designed for dynamic experiments. *bioRxiv* [Preprint]. doi: 10.1101/2020.07.03.186387

Attard-Johnson, J., Ciardha, C. Ó, and Bindemann, M. (2019). Comparing methods for the analysis of pupillary response. *Behav. Res. Methods* 51, 83–95. doi: 10.3758/s13428-018-1108-6

Barrionuevo, P. A., McAnany, J. J., Zele, A. J., and Cao, D. (2018). Non-linearities in the rod and cone photoreceptor inputs to the afferent pupil light response. *Front. Neurol.* 9:1140. doi: 10.3389/fneur.2018.01140

Barten, P. G. (1999). "Contrast sensitivity of the human eye and its effects on image quality," in *Proceedings of the Contrast Sensit. Hum. Eye Its Eff. Image Qual*, 27–29. doi: 10.1117/3.353254

Beatty, J. (1982). Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychol. Bull.* 91, 276–292. doi: 10.1037/0033-2909.91.2.276

Beatty, J., and Wagoner, B. L. (1978). Pupillometric signs of brain activation vary with level of cognitive processing. *Science* 199, 1216–1218. doi: 10.1126/science.628837

Berman, S. M., Jewett, D. L., Fein, G., Saika, G., and Ashford, F. (1990). Photopic luminance does not always predict perceived room brightness. *Light. Res. Technol.* 22, 37–41. doi: 10.1177/096032719002200103

Berson, D. M. (2003). Strange vision: ganglion cells as circadian photoreceptors. *Trends Neurosci.* 26, 314–320. doi: 10.1016/S0166-2236(03)00130-9

Berson, D. M., Dunn, F. A., and Takao, M. (2002). Phototransduction by retinal ganglion cells that set the circadian clock. *Science* 295, 1070–1073. doi: 10.1126/science.1067262

Besenecker, U. C., and Bullough, J. D. (2017). Investigating visual mechanisms underlying scene brightness. *Light. Res. Technol.* 49, 16–32. doi: 10.1177/1477153516628168

Binda, P., and Gamlin, P. D. (2017). Renewed attention on the pupil light reflex. *Trends Neurosci.* 40, 455–457. doi: 10.1016/j.tins.2017.06.007

Blackie, C. A., and Howland, H. C. (1999). An extension of an accommodation and convergence model of emmetropization to include the effects of illumination intensity. *Ophthalmic Physiol. Opt.* 19, 112–125. doi: 10.1016/S0275-5408(98)00077-5

Bodmann, H. W. (1992). Elements of photometry, brightness and visibility. *Light. Res. Technol.* 24, 29–42. doi: 10.1177/096032719202400104

Bombeke, K., Duthoo, W., Mueller, S. C., Hopf, J. M., and Boehler, C. N. (2016). Pupil size directly modulates the feedforward response in human primary visual cortex independently of attention. *Neuroimage* 127, 67–73. doi: 10.1016/j.neuroimage.2015.11.072

Bonmati-Carrion, M. A., Hild, K., Isherwood, C., Sweeney, S. J., Revell, V. L., Skene, D. J., et al. (2016). Relationship between human pupillary light reflex and circadian system status. *PLoS One* 11:e0162476. doi: 10.1371/journal.pone.0162476

Brainard, G. C., Hanifin, J. R., Greeson, J. M., Byrne, B., Glickman, G., Gerner, E., et al. (2001). Action spectrum for melatonin regulation in humans: evidence for a novel circadian photoreceptor. *J. Neurosci.* 21, 6405–6412. doi: 10.1523/jneurosci.21-16-06405.2001

Brisson, J., Mainville, M., Mailloux, D., Beaulieu, C., Serres, J., and Sirois, S. (2013). Pupil diameter measurement errors as a function of gaze direction in corneal reflection eyetrackers. *Behav. Res. Methods* 45, 1322–1331. doi: 10.3758/s13428-013-0327-0

Campbell, F. W. (1957). The depth of field of the human eye. *Opt. Acta Int. J. Opt.* 4, 157–164. doi: 10.1080/713826091

Campbell, F. W., and Gubisch, R. W. (1966). Optical quality of the human eye. *J. Physiol.* 186, 558–578. doi: 10.1113/jphysiol.1966.sp008056

Canver, M. C., Canver, A. C., Revere, K. E., Amado, D., Bennett, J., and Chung, D. C. (2014). Novel mathematical algorithm for pupillometric data analysis. *Comput. Methods Programs Biomed.* 113, 221–225. doi: 10.1016/j.cmpb.2013.08.008

Carle, C. F., James, A. C., Rosli, Y., and Maddess, T. (2019). Localization of neuronal gain control in the pupillary response. *Front. Neurol.* 10:203. doi: 10.3389/fneur.2019.00203

Chen, S., and Epps, J. (2014). Efficient and robust pupil size and blink estimation from near-field video sequences for human-machine interaction. *IEEE Trans. Cybern.* 44, 2356–2367. doi: 10.1109/TCYB.2014.2306916

Cherng, Y.-G., Baird, T., Chen, J.-T., and Wang, C.-A. (2020). Background luminance effects on pupil size associated with emotion and saccade preparation. *Sci. Rep.* 10:15718. doi: 10.1038/s41598-020-72954-z

Chougule, P. S., Najjar, R. P., Finkelstein, M. T., Kandiah, N., and Milea, D. (2019). Light-induced pupillary responses in Alzheimer's disease. *Front. Neurol.* 10:360. doi: 10.3389/fneur.2019.00360

CIE (2011). *CIE:200:2001: Supplementary System of Photometry*. Available online at: http://cie.co.at/publications/cie-supplementary-system-photometry (accessed July 23, 2020).

Clewett, D., Gasser, C., and Davachi, L. (2020). Pupil-linked arousal signals track the temporal organization of events in memory. *Nat. Commun.* 11:4007. doi: 10.1038/s41467-020-17851-9

Connelly, M. A., Brown, J. T., Kearns, G. L., Anderson, R. A., St Peter, S. D., and Neville, K. A. (2014). Pupillometry: a non-invasive technique for pain assessment in paediatric patients. *Arch. Dis. Child.* 99, 1125–1131. doi: 10.1136/archdischild-2014-306286

Coyne, J. T., Brown, N., Foroughi, C. K., and Sibley, C. M. (2019). Improving pupil diameter measurement accuracy in a remote eye tracking system. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 63, 49–53. doi: 10.1177/1071181319631176

Crawford, B. H. (1936). The dependence of pupil size upon external light stimulus under static and variable conditions. *Proc. R. Soc. London. Ser. B Biol. Sci.* 121, 376–395. doi: 10.1098/rspb.1936.0072

Crippa, S. V., Domellöf, F. P., and Kawasaki, A. (2018). Chromatic pupillometry in children. *Front. Neurol.* 9:669. doi: 10.3389/fneur.2018.00669

de Groot, S. G., and Gebhard, J. W. (1952). Pupil size as determined by adapting luminance. *J. Opt. Soc. Am.* 42:492. doi: 10.1364/JOSA.42.000492

de Winter, J. C. F., Petermeijer, S. M., Kooijman, L., and Dodou, D. (2021). Replicating five pupillometry studies of Eckhard Hess. *Int. J. Psychophysiol* 165, 145–205. doi: 10.1016/j.ijpsycho.2021.03.003

Dey, S., and Samanta, D. (2007). "An efficient approach for pupil detection in iris images," in *Proceedings of the 15th Int. Conf. Adv. Comput. Commun. ADCOM 2007*, Guwahati, 382–387. doi: 10.1109/adcom.2007.79

Do, M. T. H. (2019). Melanopsin and the intrinsically photosensitive retinal ganglion cells: biophysics to behavior. *Neuron* 104, 205–226. doi: 10.1016/j.neuron.2019.07.016

Do, M. T. H., Kang, S. H., Xue, T., Zhong, H., Liao, H. W., Bergles, D. E., et al. (2009). Photon capture and signalling by melanopsin retinal ganglion cells. *Nature* 457, 281–287. doi: 10.1038/nature07682

Ebisawa, Y. (1994). "Improved video-based eye-gaze detection method," in *Proceedings of the Conf. Proc. - 10th Anniv. IMTC 1994 Adv. Technol. I M. 1994 IEEE Instrum. Meas. Technol. Conf*, Hamamatsu, 963–966. doi: 10.1109/IMTC.1994.351964

Ebisawa, Y. (2004). "Realtime 3D position detection of human pupil," in *Proceedings of the 2004 IEEE Symp. Virtual Environ. Human-Computer Interfaces Meas. Syst. VECIMS*, Boston, MA, 8–12. doi: 10.1109/vecims.2004.1397176

Ecker, J. L., Dumitrescu, O. N., Wong, K. Y., Alam, N. M., Chen, S.-K., LeGates, T., et al. (2010). Melanopsin-expressing retinal ganglion-cell photoreceptors: cellular diversity and role in pattern vision. *Neuron* 67, 49–60. doi: 10.1016/j.neuron.2010.05.023

Eivazi, S., Santini, T., Keshavarzi, A., Kübler, T., and Mazzei, A. (2019). "Improving real-time CNN-based pupil detection through domain-specific data augmentation," in *Proceedings of the Eye Tracking Research and Applications Symposium (ETRA)*, (New York, NY: Association for Computing Machinery), 1–6. doi: 10.1145/3314111.3319914

Freedman, M. S. (1999). Regulation of mammalian circadian behavior by non-rod, non-cone, ocular photoreceptors. *Science* 284, 502–504. doi: 10.1126/science.284.5413.502

Fuhl, W., Eivazi, S., Hosp, B., Eivazi, A., Rosenstiel, W., and Kasneci, E. (2018a). "BORE: boosted-oriented edge optimization for robust, real time remote pupil center detection," in *Proceedings of the Eye Tracking Research and Applications Symposium (ETRA)*, (New York, NY: Association for Computing Machinery), 1–5. doi: 10.1145/3204493.3204558

Fuhl, W., Geisler, D., Santini, T., Appel, T., Rosenstiel, W., and Kasneci, E. (2018b). "CBF: circular binary features for robust and real-time pupil center detection," in *Proceedings of the Eye Tracking Research and Applications Symposium (ETRA)*, (New York, NY: Association for Computing Machinery), 1–6. doi: 10.1145/3204493.3204559

Fuhl, W., Kübler, T., Sippel, K., Rosenstiel, W., and Kasneci, E. (2015). "ExCuSe: robust pupil detection in real-world scenarios," in *Computer Analysis of Images and Patterns*, eds G. Azzopardi and N. Petkov (Cham: Springer), 39–51. doi: 10.1007/978-3-319-23192-1_4

Fuhl, W., Santini, T., Kasneci, G., and Kasneci, E. (2016b). *PupilNet: Convolutional Neural Networks for Robust Pupil Detection*. Available online at: http://arxiv.org/abs/1601.04902 (accessed October 6, 2020).

Fuhl, W., Santini, T., Kasneci, G., Rosenstiel, W., and Kasneci, E. (2017). *PupilNet v2.0: Convolutional Neural Networks for CPU based real time Robust*

Pupil Detection. Available online at: http://arxiv.org/abs/1711.00112 (accessed October 6, 2020).

Fuhl, W., Santini, T. C., Kübler, T., and Kasneci, E. (2016a). ElSe: ellipse selection for robust pupil detection in real-world environments. *Eye Track. Res. Appl. Symp.* 14, 123–130. doi: 10.1145/2857491.2857505

Gagl, B., Hawelka, S., and Hutzler, F. (2011). Systematic influence of gaze position on pupil size measurement: analysis and correction. *Behav. Res. Methods* 43, 1171–1181. doi: 10.3758/s13428-011-0109-5

Gamlin, P. D. R., McDougal, D. H., Pokorny, J., Smith, V. C., Yau, K.-W., and Dacey, D. M. (2007). Human and macaque pupil responses driven by melanopsin-containing retinal ganglion cells. *Vision Res.* 47, 946–954. doi: 10.1016/j.visres.2006.12.015

Goñi, S., Echeto, J., Villanueva, A., and Cabeza, R. (2004). Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system. *Proc. Int. Conf. Pattern Recognit.* 4, 941–944. doi: 10.1109/ICPR.2004.1333928

Gooley, J. J., Lu, J., Chou, T. C., Scammell, T. E., and Saper, C. B. (2001). Melanopsin in cells of origin of the retinohypothalamic tract. *Nat. Neurosci.* 4:1165. doi: 10.1038/nn768

Granholm, E. L., Panizzon, M. S., Elman, J. A., Jak, A. J., Hauger, R. L., Bondi, M. W., et al. (2017). Pupillary responses as a biomarker of early risk for Alzheimer's disease. *J. Alzheimer's Dis.* 56, 1419–1428. doi: 10.3233/JAD-161078

Güler, A. D., Ecker, J. L., Lall, G. S., Haq, S., Altimus, C. M., Liao, H. W., et al. (2008). Melanopsin cells are the principal conduits for rod-cone input to non-image-forming vision. *Nature* 453, 102–105. doi: 10.1038/nature06829

Hattar, S. (2002). Melanopsin-containing retinal ganglion cells: architecture, projections, and intrinsic photosensitivity. *Science* 295, 1065–1070. doi: 10.1126/science.1069609

Hattar, S., Kumar, M., Park, A., Tong, P., Tung, J., Yau, K.-W., et al. (2006). Central projections of melanopsin-expressing retinal ganglion cells in the mouse. *J. Comp. Neurol.* 497, 326–349. doi: 10.1002/cne.20970

Hattar, S., Lucas, R. J., Mrosovsky, N., Thompson, S., Douglas, R. H., Hankins, M. W., et al. (2003). Melanopsin and rod—cone photoreceptive systems account for all major accessory visual functions in mice. *Nature* 424, 76–81. doi: 10.1038/nature01761

Hayes, T. R., and Petrov, A. A. (2016). Mapping and correcting the influence of gaze position on pupil size measurements. *Behav. Res. Methods* 48, 510–527. doi: 10.3758/s13428-015-0588-x

Hermans, S., Smet, K. A. G., and Hanselaer, P. (2018). Brightness model for neutral self-luminous stimuli and backgrounds. *LEUKOS J. Illum. Eng. Soc. North Am.* 14, 231–244. doi: 10.1080/15502724.2018.1448280

Hiley, J. B., Redekopp, A. H., and Fazel-Rezai, R. (2006). A low cost human computer interface based on eye tracking. *Annu. Int. Conf. IEEE Eng. Med. Biol. Proc.* 2006, 3226–3229. doi: 10.1109/IEMBS.2006.260774

Holladay, L. L. (1926). The fundamentals of glare and visibility. *J. Opt. Soc. Am.* 12:271. doi: 10.1364/JOSA.12.000271

Holmqvist, K., Nyström, M., and Mulvey, F. (2012). "Eye tracker data quality: what it is and how to measure it," in *Proceedings of the Symposium on Eye Tracking Research and Applications ETRA '12*, (New York, NY: ACM Press), 45. doi: 10.1145/2168556.2168563

Hosp, B., Eivazi, S., Maurer, M., Fuhl, W., Geisler, D., and Kasneci, E. (2020). RemoteEye: an open-source high-speed remote eye tracker: Implementation insights of a pupil- and glint-detection algorithm for high-speed remote eye tracking. *Behav. Res. Methods* 52, 1387–1401. doi: 10.3758/s13428-019-01305-2

Hreidarsson, A. B. (1982). Pupil size in insulin-dependent diabetes. Relationship to duration, metabolic control, and long-term manifestations. *Diabetes* 31, 442–448. doi: 10.2337/diab.31.5.442

Hu, X., Hisakata, R., and Kaneko, H. (2019). Effects of spatial frequency and attention on pupillary response. *J. Opt. Soc. Am. A* 36:1699. doi: 10.1364/josaa.36.001699

Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C., and Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Trans. Syst. Man Cybern.* 19, 1527–1534. doi: 10.1109/21.44068

Javadi, A. H., Hakimi, Z., Barati, M., Walsh, V., and Tcheang, L. (2015). Set: a pupil detection method using sinusoidal approximation. *Front. Neuroeng.* 8:4. doi: 10.3389/fneng.2015.00004

Jennings, B. J., and Martinovic, J. (2014). Luminance and color inputs to mid-level and high-level vision. *J. Vis.* 14, 1–17. doi: 10.1167/14.2.9

Jepma, M., and Nieuwenhuis, S. (2011). Pupil diameter predicts changes in the exploration-exploitation trade-off: evidence for the adaptive gain theory. *J. Cogn. Neurosci.* 23, 1587–1596. doi: 10.1162/jocn.2010.21548

Joshi, S. (2021). Pupillometry: arousal state or state of mind? *Curr. Biol.* 31, R32–R34. doi: 10.1016/j.cub.2020.11.001

Joyce, D. S., Feigl, B., Kerr, G., Roeder, L., and Zele, A. J. (2018). Melanopsin-mediated pupil function is impaired in Parkinson's disease. *Sci. Rep.* 8:7796. doi: 10.1038/s41598-018-26078-0

Kassner, M., Patera, W., and Bulling, A. (2014). "Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction," in *Proceedings of the UbiComp 2014 - Adjun. Proc. 2014 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput*, Seattle WA, 1151–1160. doi: 10.1145/2638728.2641695

Keil, A., Albuquerque, G., Berger, K., and Magnor, M. A. (2010). "Real-time gaze tracking with a consumer-grade video camera," in *Proceedings of the 18th Int. Conf. Cent. Eur. Comput. Graph. Vis. Comput. Vision, WSCG 2010 - Co-operation with EUROGRAPHICS, Full Pap. Proc*, Plzen, Czech Republic. 129–134.

Kelbsch, C., Strasser, T., Chen, Y., Feigl, B., Gamlin, P. D., Kardon, R., et al. (2019). Standards in pupillography. *Front. Neurol.* 10:129. doi: 10.3389/fneur.2019.00129

Kercher, C., Azinfar, L., Dinalankara, D. M. R., Takahashi, T. N., Miles, J. H., and Yao, G. (2020). A longitudinal study of pupillary light reflex in 6- to 24-month children. *Sci. Rep.* 10:1205. doi: 10.1038/s41598-020-58254-6

Klingner, J. (2010). "The pupillometric precision of a remote video eye tracker," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010*, Austin, TX, 259–262. doi: 10.1145/1743666.1743727

Kobashi, H., Kamiya, K., Ishikawa, H., Goseki, T., and Shimizu, K. (2012). Daytime variations in pupil size under photopic conditions. *Optom. Vis. Sci.* 89, 197–202. doi: 10.1097/OPX.0b013e31824048a9

Kret, M. E., and Sjak-Shie, E. E. (2019). Preprocessing pupil size data: guidelines and code. *Behav. Res. Methods* 51, 1336–1342. doi: 10.3758/s13428-018-1075-y

Kumar, N., Kohlbecher, S., and Schneider, E. (2009). "A novel approach to video-based pupil tracking," in *Proceedings of the Conf. Proc. IEEE Int. Conf. Syst. Man Cybern*, San Antonio, TX, 1255–1262. doi: 10.1109/ICSMC.2009.5345909

La Morgia, C., Carelli, V., and Carbonelli, M. (2018). Melanopsin retinal ganglion cells and pupil: clinical implications for neuro-ophthalmology. *Front. Neurol.* 9:1047. doi: 10.3389/fneur.2018.01047

Lanatà, A., Armato, A., Valenza, G., and Scilingo, E. P. (2011). "Eye tracking and pupil size variation as response to affective stimuli: a preliminary study," in *Proceedings of the 2011 5th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth* Dublin, 78–84. doi: 10.4108/icst.pervasivehealth.2011.246056

Lee, J. W., Cho, C. W., Shin, K. Y., Lee, E. C., and Park, K. R. (2012). 3D gaze tracking method using Purkinje images on eye optical model and pupil. *Opt. Lasers Eng.* 50, 736–751. doi: 10.1016/j.optlaseng.2011.12.001

Lemercier, A., Guillot, G., Courcoux, P., Garrel, C., Baccino, T., and Schlich, P. (2014). Pupillometry of taste: methodological guide – from acquisition to data processing - and toolbox for MATLAB. *Quant. Methods Psychol.* 10, 179–195. doi: 10.20982/tqmp.10.2.p179

Lennie, P., Pokorny, J., and Smith, V. C. (1993). Luminance. *J. Opt. Soc. Am. A* 10:1283. doi: 10.1364/JOSAA.10.001283

Li, D., Winfield, D., and Parkhurst, D. J. (2005). "Starburst: a hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, (San Diego, CA: IEEE), 79. doi: 10.1109/CVPR.2005.531

Li, J., Li, S., Chen, T., and Liu, Y. (2018). A geometry-appearance-based pupil detection method for near-infrared head-mounted cameras. *IEEE Access* 6, 23242–23252. doi: 10.1109/ACCESS.2018.2828400

Lim, J. K. H., Li, Q. X., He, Z., Vingrys, A. J., Wong, V. H. Y., Currier, N., et al. (2016). The eye as a biomarker for Alzheimer's disease. *Front. Neurosci.* 10:536. doi: 10.3389/fnins.2016.00536

Lin, L., Pan, L., Wei, L. F., and Yu, L. (2010). "A robust and accurate detection of pupil images," in *Proceedings of the - 2010 3rd Int. Conf. Biomed. Eng. Informatics, BMEI 2010*, Yantai, 70–74. doi: 10.1109/BMEI.2010.5639646

Lin, X., Craig, J., Dean, S., Klette, G., and Klette, R. (2003). *Accurately Measuring the Size of the Pupil of the Eye*. Auckland: CITR, The University of Auckland.

Long, X., Tonguz, O. K., and Kiderman, A. (2007). "A high speed eye tracking system with robust pupil center estimation algorithm," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology*, Lyon, 3331–3334. doi: 10.1109/IEMBS.2007.4353043

Lucas, R. J., Allen, A. E., Milosavljevic, N., Storchi, R., and Woelders, T. (2020). Can We See with Melanopsin? *Annu. Rev. Vis. Sci.* 6, 453–468. doi: 10.1146/annurev-vision-030320-041239

Lucas, R. J., Douglas, R. H., and Foster, R. G. (2001). Characterization of an ocular photopigment capable of driving pupillary constriction in mice. *Nat. Neurosci.* 4, 621–626. doi: 10.1038/88443

Lucas, R. J., Peirson, S. N., Berson, D. M., Brown, T. M., Cooper, H. M., Czeisler, C. A., et al. (2014). Measuring and using light in the melanopsin age. *Trends Neurosci.* 37, 1–9. doi: 10.1016/j.tins.2013.10.004

Maclean, H., and Dhillon, B. (1993). Pupil cycle time and human immunodeficiency virus (hiv) infection. *Eye* 7, 785–786. doi: 10.1038/eye.1993.184

Manuri, F., Sanna, A., and Petrucci, C. P. (2020). PDIF: pupil detection after isolation and fitting. *IEEE Access* 8, 30826–30837. doi: 10.1109/ACCESS.2020.2973005

Martinikorena, I., Cabeza, R., Villanueva, A., Urtasun, I., and Larumbe, A. (2018). Fast and robust ellipse detection algorithm for head-mounted eye tracking systems. *Mach. Vis. Appl.* 29, 845–860. doi: 10.1007/s00138-018-0940-0

Mazziotti, R., Carrara, F., Viglione, A., Lupori, L., Lo Verde, L., Benedetto, A., et al. (2021). MEYE: web-app for translational and real-time pupillometry. *bioRxiv* [Preprint]. doi: 10.1101/2021.03.09.434438

Merritt, S. L., Schnyders, H. C., Patel, M., Basner, R. C., and O'Neill, W. (2004). Pupil staging and EEG measurement of sleepiness. *Int. J. Psychophysiol.* 52, 97–112. doi: 10.1016/j.ijpsycho.2003.12.007

Moon, P., and Spencer, D. E. (1944). On the stiles-crawford effect. *J. Opt. Soc. Am.* 34:319. doi: 10.1364/JOSA.34.000319

Morad, Y., Lemberg, H., Yofe, N., and Dagan, Y. (2000). Pupillography as an objective indicator of fatigue. *Curr. Eye Res.* 21, 535–542. doi: 10.1076/0271-3683(200007)2111-ZFT535

Morimoto, C. H., Amir, A., and Flickner, M. (2002). "Detecting eye position and gaze from a single camera and 2 light sources," in *Proceedings of the International Conference on Pattern Recognition*, Quebec City, QC, 314–317. doi: 10.1109/icpr.2002.1047459

Morimoto, C. H., Koons, D., Amir, A., and Flickner, M. (2000). Pupil detection and tracking using multiple light sources. *Image Vis. Comput.* 18, 331–335. doi: 10.1016/S0262-8856(99)00053-0

Münch, M., Léon, L., Crippa, S. V., and Kawasaki, A. (2012). Circadian and wake-dependent effects on the pupil light reflex in response to narrow-bandwidth light pulses. *Investig. Ophthalmol. Vis. Sci.* 53, 4546–4555. doi: 10.1167/iovs.12-9494

Mure, L. S. (2021). Intrinsically photosensitive retinal ganglion cells of the human retina. *Front. Neurol.* 12: 636330. doi: 10.3389/fneur.2021.636330

Murphy, P. R., Vandekerckhove, J., and Nieuwenhuis, S. (2014). Pupil-linked arousal determines variability in perceptual decision making. *PLoS Comput. Biol.* 10: e1003854. doi: 10.1371/journal.pcbi.1003854

Murray, I. J., Kremers, J., McKeefry, D., and Parry, N. R. A. (2018). Paradoxical pupil responses to isolated M-cone increments. *J. Opt. Soc. Am. A* 35:B66. doi: 10.1364/josaa.35.000b66

Murray, N. P., Hunfalvay, M., and Bolte, T. (2017). The reliability, validity, and normative data of interpupillary distance and pupil diameter using eye-tracking technology. *Transl. Vis. Sci. Technol.* 6:2. doi: 10.1167/tvst.6.4.2

OpenCV (2020). *OpenCV: Camera Calibration and 3D Reconstruction*. Available online at: https://docs.opencv.org/master/d9/d0c/group__calib3d.html (accessed November 16, 2020).

Ostrin, L. A., Abbott, K. S., and Queener, H. M. (2017). Attenuation of short wavelengths alters sleep and the ipRGC pupil response. *Ophthalmic Physiol. Opt.* 37, 440–450. doi: 10.1111/opo.12385

Pedrotti, M., Lei, S., Dzaack, J., and Rötting, M. (2011). A data-driven algorithm for offline pupil signal preprocessing and eyeblink detection in low-speed eye-tracking protocols. *Behav. Res. Methods* 43, 372–383. doi: 10.3758/s13428-010-0055-7

Pedrotti, M., Mirzaei, M. A., Tedesco, A., Chardonnet, J. R., Mérienne, F., Benedetto, S., et al. (2014). Automatic stress classification with pupil diameter analysis. *Int. J. Hum. Comput. Interact.* 30, 220–236. doi: 10.1080/10447318.2013.848320

Pérez, A., Córdoba, M. L., García, A., Méndez, R., Muñoz, M. L., Pedraza, J. L., et al. (2003). *A Precise Eye-Gaze Detection and Tracking System*. Plzen, Czech Republic.

Pinheiro, H. M., and da Costa, R. M. (2021). Pupillary light reflex as a diagnostic aid from computational viewpoint: a systematic literature review. *J. Biomed. Inform.* 117:103757. doi: 10.1016/j.jbi.2021.103757

Provencio, I., Jiang, G., De Grip, W. J., Pär Hayes, W., and Rollag, M. D. (1998). Melanopsin: an opsin in melanophores, brain, and eye. *Proc. Natl. Acad. Sci. U.S.A.* 95, 340–345. doi: 10.1073/pnas.95.1.340

Provencio, I., Rodriguez, I. R., Jiang, G., Hayes, W. P., Moreira, E. F., and Rollag, M. D. (2000). A novel human opsin in the inner retina. *J. Neurosci.* 20, 600–605. doi: 10.1523/JNEUROSCI.20-02-00600.2000

Rao, F., Chan, A. H. S., and Zhu, X. F. (2017). Effects of photopic and cirtopic illumination on steady state pupil sizes. *Vision Res.* 137, 24–28. doi: 10.1016/j.visres.2017.02.010

Rea, M. S., and Figueiro, M. G. (2018). Light as a circadian stimulus for architectural lighting. *Light. Res. Technol.* 50, 497–510. doi: 10.1177/1477153516682368

Reeves, P. (1918). Rate of pupillary dilation and contraction. *Psychol. Rev.* 25, 330–340. doi: 10.1037/h0075293

Rote, G. (1991). Computing the minimum Hausdorff distance between two point sets on a line under translation. *Inf. Process. Lett.* 38, 123–127. doi: 10.1016/0020-0190(91)90233-8

Ruby, N. F., Brennan, T. J., Xie, X., Cao, V., Franken, P., Heller, H. C., et al. (2002). Role of melanopsin in circadian responses to light. *Science* 298, 2211–2213. doi: 10.1126/science.1076701

Rukmini, A. V., Milea, D., Aung, T., and Gooley, J. J. (2017). Pupillary responses to short-wavelength light are preserved in aging. *Sci. Rep.* 7, 1–9. doi: 10.1038/srep43832

Sagawa, K. (2006). Toward a CIE supplementary system of photometry: brightness at any level including mesopic vision. *Ophthalmic Physiol. Opt.* 26, 240–245. doi: 10.1111/j.1475-1313.2006.00357.x

San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., et al. (2010). "Evaluation of a low-cost open-source gaze tracker," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, Austin, TX, 77–80. doi: 10.1145/1743666.1743685

Santini, T., Fuhl, W., Geisler, D., and Kasneci, E. (2017). "EyeRecToo: open-source software for real-time pervasive head-mounted eye tracking," in *VISIGRAPP 2017 Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, (Setúbal: SciTePress), 96–101. doi: 10.5220/0006224700960101

Santini, T., Fuhl, W., and Kasneci, E. (2018a). PuRe: robust pupil detection for real-time pervasive eye tracking. *Comput. Vis. Image Underst.* 170, 40–50. doi: 10.1016/j.cviu.2018.02.002

Santini, T., Fuhl, W., and Kasneci, E. (2018b). "PuReST: robust pupil tracking for real-time pervasive eye tracking," in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA), 2018*, Warsaw. doi: 10.1145/3204493.3204578

Schluroff, M., Zimmermann, T. E., Freeman, R. B., Hofmeister, K., Lorscheid, T., and Weber, A. (1986). Pupillary responses to syntactic ambiguity of sentences. *Brain Lang.* 27, 322–344. doi: 10.1016/0093-934X(86)90023-4

Schmidt, T. M., Alam, N. M., Chen, S., Kofuji, P., Li, W., Prusky, G. T., et al. (2014). A role for melanopsin in alpha retinal ganglion cells and contrast detection. *Neuron* 82, 781–788. doi: 10.1016/j.neuron.2014.03.022

Schneider, M., Elbau, I. G., Nantawisarakul, T., Pöhlchen, D., Brückl, T., BeCOME Working, et al. (2020). Pupil dilation during reward anticipation is correlated to depressive symptom load in patients with major depressive disorder. *Brain Sci.* 10:906. doi: 10.3390/brainsci10120906

Schwalm, M., and Jubal, E. R. (2017). Back to pupillometry: how cortical network state fluctuations tracked by pupil dynamics could explain neural signal variability in human cognitive neuroscience. *eNeuro* 4: ENEURO.0293-16.2017. doi: 10.1523/ENEURO.0293-16.2017

Schwarz, L., Gamba, H. R., Pacheco, F. C., Ramos, R. B., and Sovierzoski, M. A. (2012). "Pupil and iris detection in dynamic pupillometry using the OpenCV library," in *Proceedings of the 2012 5th Int. Congr. Image Signal Process. CISP 2012*, Chongqing, 211–215. doi: 10.1109/CISP.2012.6469846

Schwiegerling, J. (2000). Theoretical limits to visual performance. *Surv. Ophthalmol.* 45, 139–146. doi: 10.1016/S0039-6257(00)00145-4

Sharpe, L. T., Stockman, A., Jagla, W., and Jaägle, H. (2005). A luminous efficiency function, V*(λ), for daylight adaptation. *J. Vis.* 5, 948–968. doi: 10.1167/5.11.3

Sibley, C., Foroughi, C. K., Brown, N. L., Phillips, H., Drollinger, S., Eagle, M., et al. (2020). More than means: characterizing individual differences in pupillary dilations. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 64, 57–61. doi: 10.1177/1071181320641017

Smith, V. C., Pokorny, J., Lee, B. B., and Dacey, D. M. (2008). Sequential processing in vision: the interaction of sensitivity regulation and temporal dynamics. *Vision Res.* 48, 2649–2656. doi: 10.1016/j.visres.2008.05.002

Solomon, S. G., and Lennie, P. (2007). The machinery of colour vision. *Nat. Rev. Neurosci.* 8, 276–286. doi: 10.1038/nrn2094

Spitschan, M. (2019a). Melanopsin contributions to non-visual and visual function. *Curr. Opin. Behav. Sci.* 30, 67–72. doi: 10.1016/j.cobeha.2019.06.004

Spitschan, M. (2019b). Photoreceptor inputs to pupil control. *J. Vis.* 19:5. doi: 10.1167/19.9.5

Spitschan, M., Lazar, R., Yetik, E., and Cajochen, C. (2019). No evidence for an S cone contribution to acute neuroendocrine and alerting responses to light. *Curr. Biol.* 29, R1297–R1298. doi: 10.1016/j.cub.2019.11.031

Stanley, P., and Davies, A. (1995). The effect of field of view size on steady-state pupil diameter. *Ophthalmic Physiol. Opt.* 15, 601–603. doi: 10.1016/0275-5408(94)00019-V

Stockman, A., and Sharpe, L. T. (2000). The spectral sensitivities of the middle- and long-wavelength-sensitive cones derived from measurements in observers of known genotype. *Vision Res.* 40, 1711–1737. doi: 10.1016/S0042-6989(00)00021-3

Świrski, L., Bulling, A., and Dodgson, N. (2012). "Robust real-time pupil tracking in highly off-axis images," in *ETRA '12: Proceedings of the Symposium on Eye Tracking Research and Applications*, Santa Barbara, CA, 173–176. doi: 10.1145/2168556.2168585

Świrski, L., and Dodgson, N. A. (2013). "A fully-automatic, temporal approach to single camera, glint-free 3D eye model fitting," in *Proceedings of the Pervasive Eye Track. Mob. Eye-Based Interact*, Lund.

Tabashum, T., Zaffer, A., Yousefzai, R., Colletta, K., Jost, M. B., Park, Y., et al. (2021). Detection of Parkinson's disease through automated pupil tracking of the post-illumination pupillary response. *Front. Med.* 8:645293. doi: 10.3389/fmed.2021.645293

Tähkämö, L., Partonen, T., and Pesonen, A. K. (2019). Systematic review of light exposure impact on human circadian rhythm. *Chronobiol. Int.* 36, 151–170. doi: 10.1080/07420528.2018.1527773

Thaler, L., Schütz, A. C., Goodale, M. A., and Gegenfurtner, K. R. (2013). What is the best fixation target? The effect of target shape on stability of fixational eye movements. *Vision Res.* 76, 31–42. doi: 10.1016/j.visres.2012.10.012

Thapan, K., Arendt, J., and Skene, D. J. (2001). An action spectrum for melatonin suppression: evidence for a novel non-rod, non-cone photoreceptor system in humans. *J. Physiol.* 535, 261–267. doi: 10.1111/j.1469-7793.2001.t01-1-00261.x

Titz, J., Scholz, A., and Sedlmeier, P. (2018). Comparing eye trackers by correlating their eye-metric data. *Behav. Res. Methods* 50, 1853–1863. doi: 10.3758/s13428-017-0954-y

Tkacz-Domb, S., and Yeshurun, Y. (2018). The size of the attentional window when measured by the pupillary response to light. *Sci. Rep.* 8, 1–7. doi: 10.1038/s41598-018-30343-7

Topal, C., Cakir, H. I., and Akinlar, C. (2017). *APPD*. Available online at: http://arxiv.org/abs/1709.06366 (accessed 28th December, 2020).

Truong, W., Zandi, B., Trinh, V. Q., and Khanh, T. Q. (2020). Circadian metric – Computation of circadian stimulus using illuminance, correlated colour temperature and colour rendering index. *Build. Environ.* 184:107146. doi: 10.1016/j.buildenv.2020.107146

Tsukahara, J. S., Harrison, T. L., and Engle, R. W. (2016). The relationship between baseline pupil size and intelligence. *Cogn. Psychol.* 91, 109–123. doi: 10.1016/j.cogpsych.2016.10.001

Van der Stoep, N., Van der Smagt, M. J., Notaro, C., Spock, Z., and Naber, M. (2021). The additive nature of the human multisensory evoked pupil response. *Sci. Rep.* 11:707. doi: 10.1038/s41598-020-80286-1

Van Egroo, M., Gaggioni, G., Cespedes-Ortiz, C., Ly, J. Q. M., and Vandewalle, G. (2019). Steady-state pupil size varies with circadian phase and sleep homeostasis in healthy young men. *Clocks Sleep* 1, 240–258. doi: 10.3390/clockssleep1020021

Van Meeteren, A. (1978). On the detective quantum efficiency of the human eye. *Vision Res.* 18, 257–267. doi: 10.1016/0042-6989(78)90160-8

van Rij, J., Hendriks, P., van Rijn, H., Baayen, R. H., and Wood, S. N. (2019). Analyzing the time course of pupillometric data. *Trends Hear.* 23, 1–22. doi: 10.1177/2331216519832483

Vera-Olmos, F. J., Pardo, E., Melero, H., and Malpica, N. (2018). DeepEye: deep convolutional network for pupil detection in real environments. *Integr. Comput. Aided. Eng.* 26, 85–95. doi: 10.3233/ICA-180584

Wang, D., Mulvey, F. B., Pelz, J. B., and Holmqvist, K. (2017). A study of artificial eyes for the measurement of precision in eye-trackers. *Behav. Res. Methods* 49, 947–959. doi: 10.3758/s13428-016-0755-8

Watson, A. B., and Yellott, J. I. (2012). A unified formula for light-adapted pupil size. *J. Vis.* 12, 1–16. doi: 10.1167/12.10.12

Wildemeersch, D., Baeten, M., Peeters, N., Saldien, V., Vercauteren, M., and Hans, G. (2018). Pupillary dilation reflex and pupillary pain index evaluation during general anaesthesia: a pilot study. *Rom. J. Anaesth. Intensive Care* 25, 19–23. doi: 10.21454/rjaic.7518.251.wil

Winn, M. B., Wendt, D., Koelewijn, T., and Kuchinsky, S. E. (2018). Best practices and advice for using pupillometry to measure listening effort: an introduction for those who want to get started. *Trends Hear.* 22:233121651880086. doi: 10.1177/2331216518800869

Withouck, M., Smet, K. A. G., Ryckaert, W. R., Pointer, M. R., Deconinck, G., Koenderink, J., et al. (2013). Brightness perception of unrelated self-luminous colors. *J. Opt. Soc. Am. A* 30:1248. doi: 10.1364/JOSAA.30.001248

Woodhouse, J. M. (1975). The effect of pupil size on grating detection at various contrast levels. *Vision Res.* 15, 645–648. doi: 10.1016/0042-6989(75)90278-3

Yiu, Y. H., Aboulatta, M., Raiser, T., Ophey, L., Flanagin, V. L., Eulenburg, P., et al. (2019). DeepVOG: open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *J. Neurosci. Methods* 324:108307. doi: 10.1016/j.jneumeth.2019.05.016

Young, R. S. L., and Kimura, E. (2008). Pupillary correlates of light-evoked melanopsin activity in humans. *Vision Res.* 48, 862–871. doi: 10.1016/j.visres.2007.12.016

Zandi, B., Eissfeldt, A., Herzog, A., and Khanh, T. Q. (2021). Melanopic limits of metamer spectral optimisation in multi-channel smart lighting systems. *Energies* 14:527. doi: 10.3390/en14030527

Zandi, B., Guo, X., Bodrogi, P., and Khanh, T. Q. (2018). "EXPERIMENTAL EVALUATION OF DIFFERENT BRIGHTNESS PERCEPTION MODELS BASED ON HUMAN PUPIL LIGHT RESPONSES," in *PROCEEDINGS OF CIE 2018 TOPICAL CONFERENCE ON SMART LIGHTING*, (Taipei: International Commission on Illumination, CIE), 201–208. doi: 10.25039/x45.2018.OP34

Zandi, B., and Khanh, T. Q. (2021). Deep learning-based pupil model predicts time and spectral dependent light responses. *Sci. Rep.* 11:841. doi: 10.1038/s41598-020-79908-5

Zandi, B., Klabes, J., and Khanh, T. Q. (2020). Prediction accuracy of L- and M-cone based human pupil light models. *Sci. Rep.* 10:10988. doi: 10.1038/s41598-020-67593-3

Zele, A. J., Adhikari, P., Cao, D., and Feigl, B. (2019). Melanopsin and cone photoreceptor inputs to the afferent pupil light response. *Front. Neurol.* 10:529. doi: 10.3389/fneur.2019.00529

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 1330–1334. doi: 10.1109/34.888718

Zhu, D., Moore, S. T., and Raphan, T. (1999). Robust pupil center detection using a curvature algorithm. *Comput. Methods Programs Biomed.* 59, 145–157. doi: 10.1016/S0169-2607(98)00105-9

Conflict of Interest: 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.

frontiers in Neuroinformatics

BRIEF RESEARCH REPORT
published: 11 August 2021
doi: 10.3389/fninf.2021.704627

Fitting Splines to Axonal Arbors Quantifies Relationship Between Branch Order and Geometry

Thomas L. Athey[1,2*], *Jacopo Teneggi*[2], *Joshua T. Vogelstein*[1,2,3,4], *Daniel J. Tward*[5,6], *Ulrich Mueller*[7] *and Michael I. Miller*[1,2,3,4]

[1] Institute for Computational Medicine, Johns Hopkins University, Baltimore, MD, United States, [2] Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD, United States, [3] Center for Imaging Science, Johns Hopkins University, Baltimore, MD, United States, [4] Kavli Neuroscience Discovery Institute, Johns Hopkins University, Baltimore, MD, United States, [5] Department of Computational Medicine, University of California, Los Angeles, Los Angeles, CA, United States, [6] Department of Neurology, University of California, Los Angeles, Los Angeles, CA, United States, [7] Department of Neuroscience, Johns Hopkins University, Baltimore, MD, United States

Neuromorphology is crucial to identifying neuronal subtypes and understanding learning. It is also implicated in neurological disease. However, standard morphological analysis focuses on macroscopic features such as branching frequency and connectivity between regions, and often neglects the internal geometry of neurons. In this work, we treat neuron trace points as a sampling of differentiable curves and fit them with a set of branching B-splines. We designed our representation with the Frenet-Serret formulas from differential geometry in mind. The Frenet-Serret formulas completely characterize smooth curves, and involve two parameters, curvature and torsion. Our representation makes it possible to compute these parameters from neuron traces in closed form. These parameters are defined continuously along the curve, in contrast to other parameters like tortuosity which depend on start and end points. We applied our method to a dataset of cortical projection neurons traced in two mouse brains, and found that the parameters are distributed differently between primary, collateral, and terminal axon branches, thus quantifying geometric differences between different components of an axonal arbor. The results agreed in both brains, further validating our representation. The code used in this work can be readily applied to neuron traces in SWC format and is available in our open-source Python package `brainlit`: http://brainlit.neurodata.io/.

Keywords: neuron, morphology, axon, curvature, projection, mouse, spline, python

OPEN ACCESS

Edited by:
William T. Katz, Janelia Research Campus, United States

Reviewed by:
Mark Kittisopikul, Janelia Research Campus, United States
Kevin Boergens, Paradromics, Inc., United States

***Correspondence:**
Thomas L. Athey
tathey1@jhu.edu

Received: *03 May 2021*
Accepted: *05 July 2021*
Published: *11 August 2021*

Citation:
Athey TL, Teneggi J, Vogelstein JT, Tward DJ, Mueller U and Miller MI (2021) Fitting Splines to Axonal Arbors Quantifies Relationship Between Branch Order and Geometry. Front. Neuroinform. 15:704627. doi: 10.3389/fninf.2021.704627

1. INTRODUCTION

Not long after scientists like Ramon y Cajal started studying the nervous system with staining and microscopy, neuron morphology became a central topic in neuroscience (Parekh and Ascoli, 2013). Morphology became the obvious way to organize neurons into categories such as pyramidal cells, Purkinje cells, and stellate cells. However, morphology is important not only for neuron subtyping, but in understanding learning and disease. For example, a now classic neuroscience experiment found altered morphology in geniculocortical axonal arbors in kittens whose eyes had been stitched shut upon birth (Antonini and Stryker, 1993). Also, morphological changes have been associated with the gene underlying an inherited form of Parkinson's disease (MacLeod et al., 2006). Neuron morphology has been an important part of neuroscience for over a century, and remains so – one of the BRAIN Initiative Cell Census Network's primary goals is to systematically characterize neuron morphology in the mammalian brain.

Currently, studying neuron morphology typically involves imaging one or more neurons, then tracing the cells and storing the traces in a digital format. Several recent initiatives have accumulated large datasets of neuron traces to facilitate morphology research. NeuroMorpho.Org, for example, hosts a total of over 140,000 neuron traces from a variety of animal species (Ascoli et al., 2007). These traces are typically stored as a list of vertices, each with some associated attributes including connections to other vertices.

Many scientists analyze neuron morphology by computing various summary features such as number of branch points, total length, and total encompassed volume. Neurolucida, a popular neuromorphology software, employs this technique. Another approach focuses on neuron topology, and uses metrics such as tree edit distance (Heumann and Wittum, 2009). However, both of these approaches neglect *kinematic* geometry, or how the neuron travels through space. Tortuosity index is a summary feature that captures internal axon geometry, but this feature depends on the definition of start and end points, and cannot capture an axon's curvature at a single point.

In this work, we look at neuron traces through the lens of differential geometry. In particular, we establish a system of fitting interpolating splines to the neuron traces, and computing their curvature and torsion properties. To our knowledge, curvature and torsion have never been measured in neuron traces. We applied this method to cortical projection neuron traces from two mouse brains in the MouseLight dataset from HHMI Janelia (Winnubst et al., 2019). In both brains, we found different distributions of these properties between primary, collateral, and terminal axon segments. The code used in this work is available in our open-source Python package `brainlit`: http://brainlit.neurodata.io/.

2. METHODS

2.1. Spline Fitting

First, the neuron traces were split into segments by recursively identifying the longest root to leaf path (**Figure 1A**). The first axon segment to be isolated in this way was defined to be the "primary" segment. Subsequent segments that branched were defined as "collateral" segments, and those that did not branch were defined to be "terminal" segments (**Figure 1B**). This classification approximates the standard morphological definitions of primary, collateral and terminal axon branches.

Next, a B-spline was fit to each point sequence using `scipy`'s function `splprep` (Virtanen et al., 2020). Kunoth et al. (2018) provide an in depth description of B-splines and their applications. Briefly, B-splines are linear combinations of piecewise polynomials, sometimes called basis functions. The basis functions are defined by a set of knots, which determine where the polynomial pieces meet, and degree, which determines the degree of the polynomial pieces. The j'th basis function for a set of knots ξ and degree p is recursively defined by Equation (1.1) in Kunoth et al. (2018):

$$B_{j,p,\xi} := \frac{x-\xi_j}{\xi_{j+p}-\xi_j} B_{j,p-1,\xi}(x) + \frac{\xi_{j+p+1}-x}{\xi_{j+p+1}-\xi_{j+1}} B_{j+1,p-1,\xi}(x)$$

with

$$B_{i,0,\xi} := \begin{cases} 1, & \text{if } x \in [\xi_i, \xi_{i+1}), \\ 0, & \text{otherwise.} \end{cases}$$

Splines are fit to data by solving a constrained optimization problem, where a smoothing term is minimized while keeping the residual error under a specified value (Dierckx, 1982). Here, we constrain the splines to pass exactly through all points in the original trace, which corresponds to a smoothing condition of $s = 0$ in `splprep`. For a sequence of $n > 5$ points, we fit a spline of degree 5, which is the minimal degree that ensures that the splines are thrice continuously differentiable. Differentiability is important because it allows for estimation of curvature and torsion, explained in the next section.

Sequences of fewer than 5 points, however, required lower degree splines to fully constrain the fitting procedure. For a sequence of $3 < n \leq 5$ points we used degree 3, for a sequence of $n = 3$ points we used degree 2, and for a sequence of $n = 2$ points we used degree 1. By selecting the degree in this way, we avoided splines of large even degree, such as fourth order splines, which are not recommended in our interpolation setting (Virtanen et al., 2020). Also, these degree choices are low enough to allow for a fully constrained fitting procedure, but high enough to make curvature/torsion nonvanishing when possible.

We recall that B-splines are not required to be parameterized by the arclength of the curve. Here, we set $\xi = \{0, \ldots, L\}$, where L is the cumulative length of the segments connecting the vertices of the trace, in μm. All other spline fitting options were set to the defaults in `splprep`. This spline fitting method can be applied to any set of points organized in a tree structure, such as a SWC file. **Figure 1C** shows examples of splines that were fit to neuron traces.

2.2. Frenet-Serret Parameters

An important advantage of B-splines is that their derivatives can be computed in closed form. In fact, their derivatives are defined in terms of B-splines as shown below in Theorem 3 from Kunoth et al. (2018):

Theorem *For a continuously differentiable b-spline $B_{j,p,\xi}(\cdot)$ defined by index j, degree $p \geq 1$, and knot sequence ξ, we have:*

$$\frac{d}{ds} B_{j,p,\xi}(s) = p\left(\frac{B_{j,p-1,\xi}(s)}{\xi_{j+p}-\xi_j} - \frac{B_{j+1,p-1,\xi}(s)}{\xi_{j+p+1}-\xi_{j+1}}\right)$$

where we assume by convention that fractions with zero denominator have value zero.

Curvature and torsion can be easily computed because of this property. For a thrice differentiable curve $x(s) \in \mathbb{R}^3$ that is parameterized by arclength (i.e., $||\dot{x}(s)|| = 1 \ \forall s$), one can

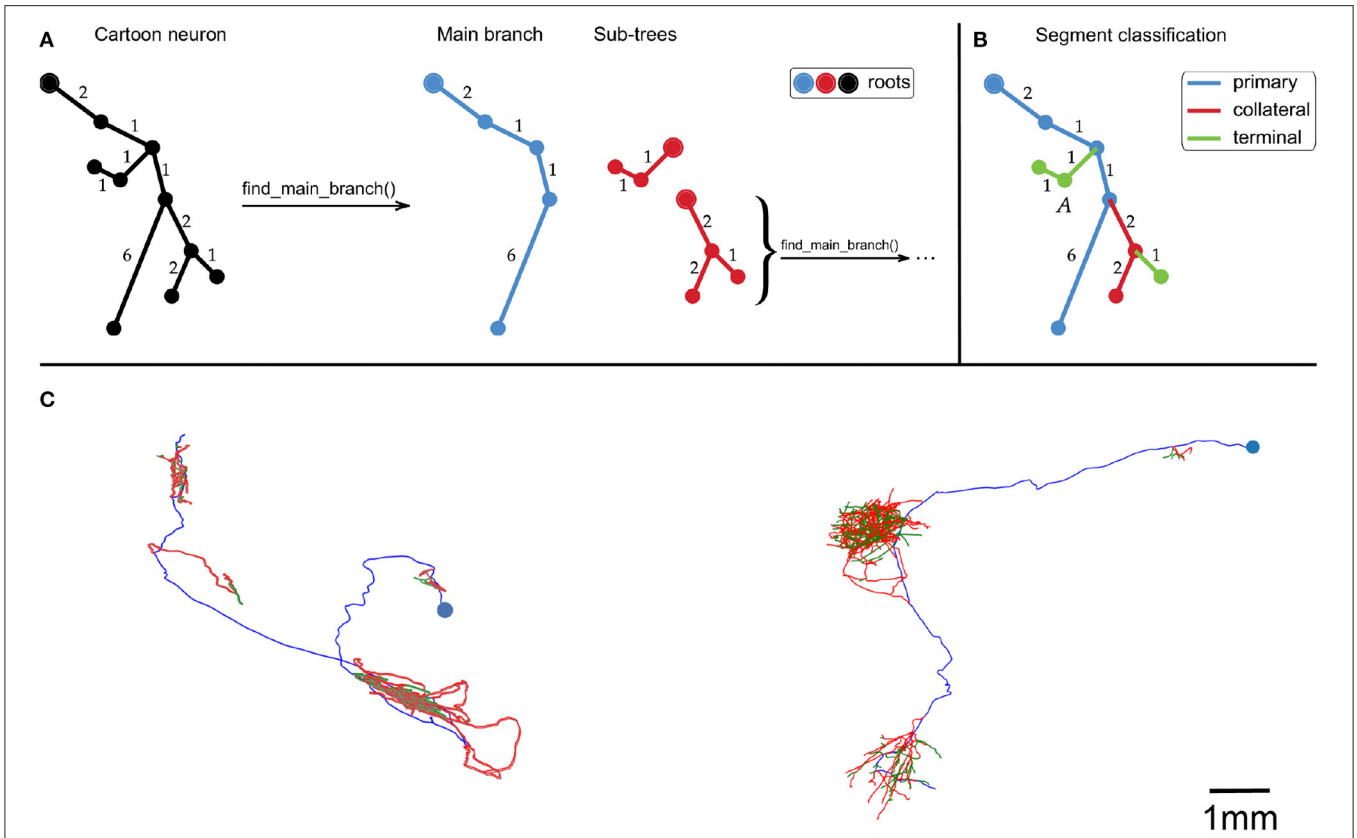


FIGURE 1 | (A,B) Cartoon example of how we partition an axonal arbor trace into different segment classes, with numbers indicating distances between points (with arbitrary units). **(A)** A neuron trace is split into different segments by identifying the longest root to leaf path ("Main branch"), and separating sub-trees from it. The sub-trees which still have branch points are processed in the same way until the neuron has been split into segments. By using path length to identify the Main branch, this splitting process is invariant to rigid transformations of the trace. **(B)** Illustration of how axon segments are classified as primary, collateral, or terminal. The first segment is defined as primary, and segments that have no sub-trees are defined as terminal. All other segments are defined as collateral. **(C)** Examples of our spline fitting method applied to neuron traces from the MouseLight project (Winnubst et al., 2019). The splines pass through all trace points, and are thrice continuously differentiable for segments that contain at least five trace points. The blue points indicate the somas, and the spline colors indicate segment class (blue = primary, red = collateral, green = terminal). The neuron on the left is from brain 1, the one on the right is from brain 2. The scale bar only applies to **(C)**.

compute the curvature (κ) and torsion (τ) with the following formulas:

$$\kappa(s) = ||\dot{x}(s) \times \ddot{x}(s)||$$
$$\tau(s) = \frac{\langle(\dot{x}(s) \times \ddot{x}(s)), \dddot{x}(s)\rangle}{||\dot{x}(s) \times \ddot{x}(s)||^2}$$

defined with the standard Euclidean norm $||\cdot||$, inner product $\langle\cdot,\cdot\rangle$, and cross product $\times$. When curvature vanishes, we define torsion to be zero as well, since the torsion equation becomes undefined. The units of curvature and torsion are both inverse length. In this work, neuron traces have units of microns, so curvature and torsion both have units of $(\mu m)^{-1}$.

Curvature measures how much a curve deviates from being straight, and torsion measures how much a curve deviates from being planar. Together, these quantities parametrize the Frenet-Serret formulas of differential geometry. These formulas completely characterize continuously differentiable curves in three-dimensional Euclidean space, up to rigid motion (Grenander et al., 2007). Curvature takes non-negative values, but torsion can be positive or negative where the sign denotes the direction of the torsion in the right-handed coordinate system. In this work, we are not interested in the direction of the torsion, so we focused on the torsion magnitude (absolute value).

2.3. Data

We applied our methods to a collection of cortical projection neuron axon traces from two mouse brains in the HHMI Janelia MouseLight dataset. The precision of the reconstructions is limited by the resolution of the original two-photon block-face images, which was $0.3\mu m \times 0.3\mu m \times 1\mu m$ (Winnubst et al., 2019). Each reconstruction is the consensus of traces by two independent annotators. Winnubst et al. (2019) showed that using two annotators per neuron produced reconstructions that are about 93.7% accurate (in terms of total axonal length). There were 180 traces from brain 1 and 50 traces from brain 2.

After fitting splines to these traces, curvature and torsion magnitude were sampled every $1\mu m$ along the axon segments. Sampling every $1\mu m$ is the highest sampling frequency that

does not exceed the image resolution, so it is an appropriate balance of precision and computational efficiency. We studied curvature and torsion magnitude in two ways, described below in sections 2.4, 2.5.

2.4. Computing Autocorrelation of Curvature and Torsion

Our first goal was to identify the length scale at which straight axon segments remain straight and curved axon segments remain curved, so we studied the autocorrelation of curvature and torsion magnitude along the axon segments. For each axon segment, the autocorrelation functions of curvature and torsion were computed along the length of the segment, yielding a collection of autocorrelation functions for each brain. Then, we evaluated whether autocorrelation at a particular lag was significantly higher than 0.3 using a one-sided t-test with a significance threshold of $\alpha = 0.05$. We identified 0.3 as our effect size because correlations higher than 0.3 are generally regarded as "moderate" correlations.

It is worth noting that, by the nature of the spline fitting procedure in Virtanen et al. (2020), "lag" in our autocorrelation functions refers to straight line distances between the trace points, not by the arclength of the resulting curves.

2.5. Comparing Axon Segment Classes

Our second goal in the analysis was to compare curvature/torsion between segment classes. First, we estimated each segment's average curvature/torsion magnitude by taking the mean from all points that were sampled on that segment.

In order to compare different segment classes, we developed a paired sample method for testing for differences in average curvature/torsion. Different neurons represented different samples, and the average curvature/torsion of two segment classes (primary vs. collateral, collateral vs. terminal, primary vs. terminal) represented the paired measurements.

Define the random variable X as the average curvature/torsion of one segment class and Y as the average curvature/torsion of another segment class. Further, say X and Y are both real valued. Our null and alternative hypotheses are as follows:

$$\mathcal{H}_0 : \Pr[X > Y] = 0.5$$
$$\mathcal{H}_1 : \Pr[X > Y] \neq 0.5$$

We tested these hypotheses using the sign test (Neuhauser, 2011). The test statistic is the number of times that the data point from one sample is greater than its pair from the other sample. A key advantage of the sign test is that it does not require parametric distribution assumptions, such as normality of the data. Also, its null distribution can be computed exactly via the binomial distribution. The two different parameters (curvature and torsion), and the three different segment class pairs constitute six total tests, so we applied the Bonferroni correction to $\alpha = 0.05$ to obtain the significance threshold 0.0083, which controls the family-wise error rate to 0.05. We conducted one-sided sign tests in all cases.

We also wanted to study whether these results would hold if the traces were perturbed. In particular, since the annotators vary the distance between points in their trace, we decided to randomly remove trace points and repeat the curvature/torsion measurements. Since the traces are tree structures, a trace point can be removed after connecting its child node(s) to its parent node. We produced 20 copies of the original dataset and, in each case, removed every trace point with 10% probability.

3. RESULTS

3.1. Autocorrelation of Curvature and Torsion

The autocorrelation functions for all segments of a brain were averaged, and they are shown in **Figure 2**. Also shown is a shaded region that represents one standard deviation of these autocorrelation functions. The t-tests described in section 2.4 were significant at lags of $1, 2, 3, 4\mu m$ for curvature in brain 1, $1, 2, 3\mu m$ for curvature in brain 2, $1, 2\mu m$ for torsion in brain 1, and $1, 2\mu m$ for torsion in brain 2.

3.2. Axon Segment Class Differences

The distributions of mean curvature and torsion are shown in **Figure 3**. Our statistical testing procedure, described in section 2.5, rejected the null hypothesis in all cases, with all $p < 5 \times 10^{-7}$. The directions of the one-sided tests were identical in both brains with:

$$\text{Curvature: Collateral} > \text{Terminal} > \text{Primary}$$
$$\text{Torsion: Collateral} > \text{Primary} > \text{Terminal}$$

When we applied the same testing procedure to the 20 datasets with trace points randomly removed, the null hypotheses were also all rejected, in the same directions, in all cases.

Neuron counts for all 36 possible curvature/torsion orderings across classes are shown in **Figure 4**. The most common ordering of curvature/torsion is exactly the same as the results of the sign test (106/180 neurons followed this ordering in brain 1, 38/50 in brain 2).

In the **Supplementary Figure 1**, we plot the curvature/torsion vs. segment length. There appear to be modest correlations between segment length and curvature/torsion values in log-log plots.

4. DISCUSSION

Our work proposes a model of neuron morphology using continuously differentiable B-splines. From these curves, it is possible to measure kinematic properties of neuronal processes, including curvature and torsion. These techniques are freely available in our open source Python package `brainlit`: http://brainlit.neurodata.io/, and more information about how to reproduce the specific results here can be found in the data availability statement.

In most contemporary neuromorphological analysis, neuron traces are regarded as piecewise linear structures, which precludes any analysis of higher order derivatives. Our spline representation makes it possible to estimate higher order derivatives and study parameters like curvature and

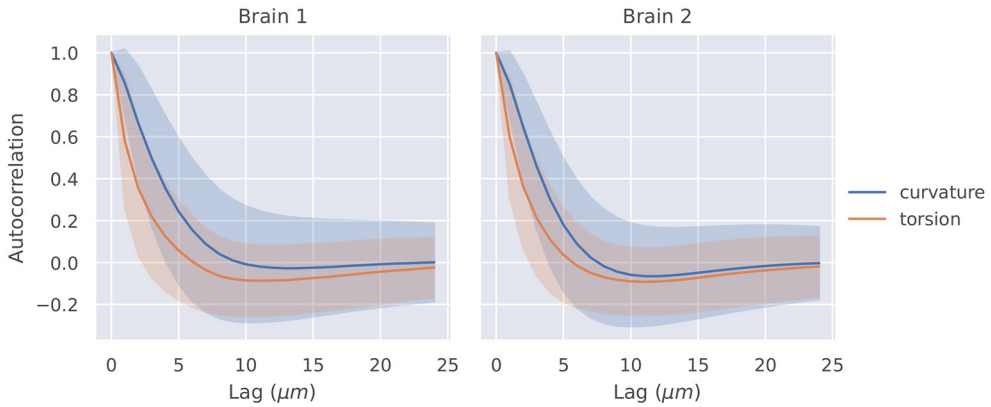


FIGURE 2 | Autocorrelation of curvature and torsion magnitude averaged across all axon segments with $\pm 1\sigma$ confidence intervals. Curvature and torsion were sampled at every $1\mu m$ along the axon segments. One sided *t*-tests indicated that curvature had statistically significant autocorrelation values above 0.3 at lags of 1, 2, 3, and 4 μm in brain 1 and 1, 2, and 3 μm in brain 2. Torsion had statistically significant autocorrelation values above 0.3 at lags of 1 and $2\mu m$ in both brain 1 and 2.

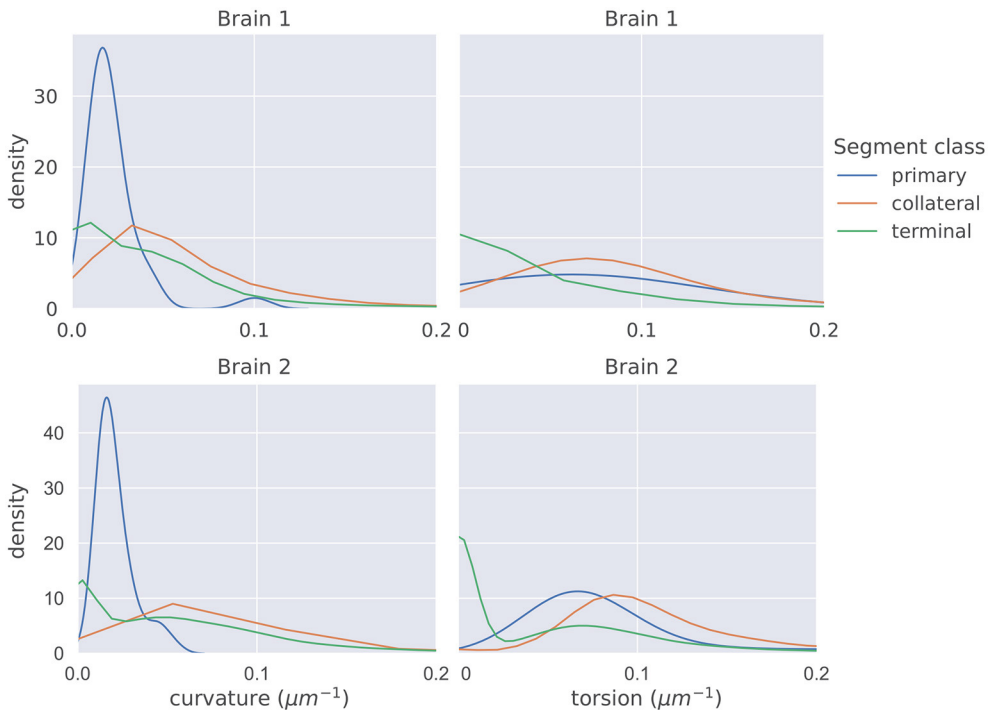


FIGURE 3 | The distributions of average curvature and average torsion differed between the different segment classes as shown in these kernel density estimates (which integrate to one, and therefore density has the units of μm), using a Gaussian kernel. The bandwidth of the kernel was 1.2σ where σ was computed using Scott's method (Scott, 2015). Segment averages were computed by sampling the curves at a uniform spacing of $1\mu m$. One-sided sign tests, testing for differences in average curvature and torsion, were conducted while controlling the family-wise error rate to 0.05. The tests were significant in all cases and the directionality of the tests agreed in both brains.

torsion of neuron branches. In the popular piecewise linear representation, curvature and torsion would be zero along the line segments, and undefined where the line segments meet. We simulated a piecewise linear representation by modifying our spline fitting procedure to only produce splines of degree one. Indeed, with this less sophisticated representation, curvature and torsion vanished everywhere, making them not meaningful.

Tortuosity index captures similar information to our curvature/torsion measurements and is popular in neuromorphological analysis (Stepanyants et al., 2004). However, tortuosity requires the user to define start and end

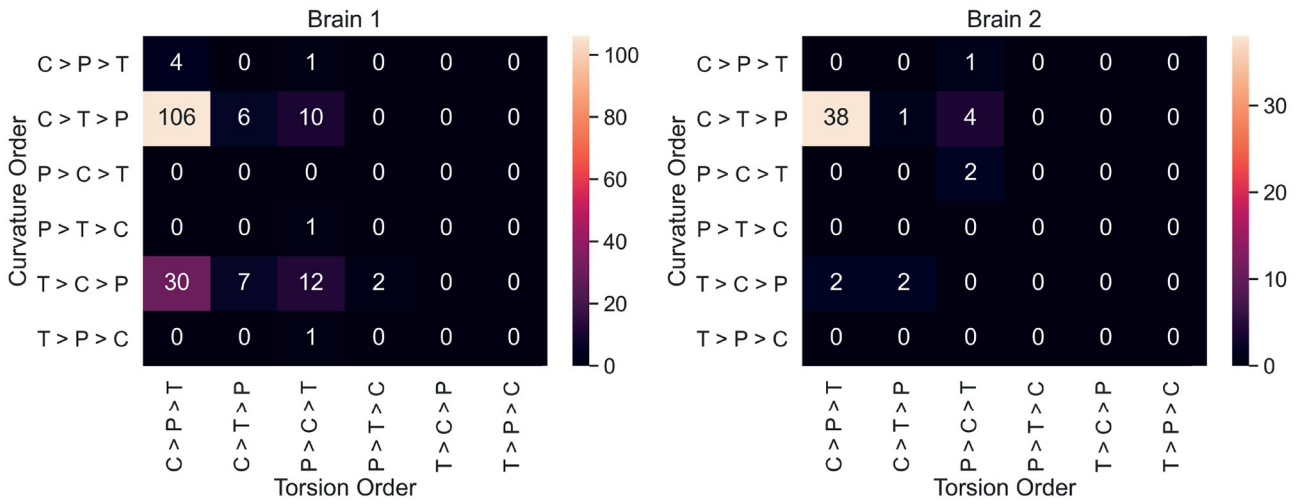


FIGURE 4 | For each neuron, average curvature and torsion was computed for all three segment classes (P, primary; C, collateral; T, terminal) and compared between classes. These heatmaps show the neuron counts for all 36 possible orderings of curvature/torsion. The most common ordering was collateral > terminal > primary for curvature and collateral > primary > terminal for torsion.

points whereas our method does not. Further, the piecewise linear representation of neuron traces limits the sampling frequency of tortuosity. Since tortuosity of a straight line is identically 1, placing the start and endpoints on the same linear segment will always produce a tortuosity value of 1. Our method, on the other hand, can produce more meaningful instantaneous curvature/torsion values.

Our methods for fitting splines and measuring curvature and torsion can be applied in neuromorphological analysis in a variety of ways, but we highlight two applications here, on a dataset of 230 projection neuron traces from two different mouse brains. We found that the autocorrelation functions of both curvature and torsion showed statistically significant correlations above 0.3 within lags of approximately 2 microns (specific lag values given in section 3.1). Next, we defined segments as either "primary," "collateral," or "terminal," and found significant differences in the distributions of curvature and torsion between these classes.

The statistical analysis approach described in section 2.5 satisfies two desirable properties. First, by averaging measurements across segment classes, and pairing the data, we did not have to assume independence between segments of the same neuron. Assuming independence seemed inappropriate because, for example, segments that are connected to each other may have correlated geometry. Second, it avoided any parametric assumptions of the data, such as assuming normality of curvature/torsion measurements. A normality assumption seemed inappropriate for several reasons, including the fact that curvature is nonnegative, and that curvature/torsion was identically 0 for short segments with only 2 trace points.

Figure 4 shows that most individual neurons agree with the overall trend that collateral segments have the highest curvature and torsion. This suggests that the finding here is a consistent phenomenon among projection neurons in mice. In order to explore curvature/torsion distributions one level deeper, we looked into the relationship between curvature/torsion and segment length (see **Supplementary Figure 1**). In all segment classes, longer segments tend to have less curvature. The relationship between segment length and torsion is weaker, but there does appear to be a positive correlation.

Together, these findings suggest that the geometry of primary axon branches is different than that of higher order branches, such as the segments in terminal arborizations. In particular, higher order branches (collaterals and terminals) had higher curvature than primary branches. Collateral branches also had the highest torsion, but primary branches had higher torsion than terminal segments.

The primary limitation of our work is that our process of splitting a neuron trace into segments may not partition an axonal arbor into the most meaningful segment classes. This is because we needed an unambiguous classification system, while most definitions used in neuroscience literature are subjective and qualitative. For example, collaterals are broadly defined as branches that split off their parent branch at sharp angles, and arborize in a different location from other branches (Rockland, 2013). However, there is no strict cutoff for how far away a branch has to travel for it to be considered a collateral. Further, a branch may be simultaneously considered a collateral and a terminal. We designed a set of segment classes which are mutually exclusive, collectively exhaustive, and agree with common usage of the terms 'primary,' 'collateral,' and 'terminal' by neuroscientists. Future work could include changing our definitions of these classes to incorporate other morphological properties such as branch angle, or axon radius. Also, extending these experiments to neuron trace repositories such as NeuroMorpho.Org would help verify if the results using our classification system generalize.

Previous research has already indicated differences in axon geometry across neuronal cell types. For example,

Stepanyants et al. (2004) found higher tortuosity in the axons of GABAergic interneurons vs. those of pyramidal cells. Similarly, Portera-Cailliau et al. (2005) found Cajal-Retzius cells to be significantly more tortuous than Thalamocortical (TC) cells, which is a type of projection neuron. Portera-Cailliau et al. (2005) also offers evidence that, while the primary axon in TC cells travel via a growth cone, most branching occurs via an interstitial, growth cone independent process. Our work elaborates on this distinction, suggesting that higher order axon branches have different geometry as well. While earlier research studied the differences of axonal geometry between neurons, we studied the variation of axonal geometry within neurons.

It is also worth noting that this is not the first work to model neuron traces as continuous curves in $\mathbb{R}^3$. For example, Duncan et al. (2018) construct a sophisticated and elegant representation of neurons that offers several useful properties. First, their representation is invariant to rigid motion and reparameterization. Second, their representation offers a vector space with a shape metric amenable to clustering and classification. However, their representation is limited to neuron topologies consisting of a main branch and only first order collaterals. Our B-splines approach does not immediately yield vector space properties, but can be applied to neurons with higher order branching, and allows for closed form computation of curvature and torsion. In short, the representation in Duncan et al. (2018) is designed for analysis between neurons, and our representation is designed for analysis within neurons. In the future, we are interested in bringing the advantages of their work to the open source software community, and combining it with the advantages of ours.

This method could also be applied to measure curvature and torsion of dendrites, since dendrites also have a tree structure and are commonly stored in SWC format. However, the segment classes that we define (primary, collateral and terminal) would be inappropriate for dendrites. A segmentation classification system for dendrites would likely depend on the neuron type being studied. For example, a natural classification system of dendrites in pyramidal cells may separate apical dendrites from basal ones while dendrites in Purkinje cells would not have such a division. The researcher would have to define the dendrite segment classes according to the dataset, and the goals of the research.

It is well known that axons are pruned and modified over time (Portera-Cailliau et al., 2005). It is possible that this process contributes to the different geometry of proximal vs. distal axonal segments. Indeed, Portera-Cailliau et al. (2005) mentions the growth of short twisted branches toward the end of axon development. Future animal experiments could follow-up on this idea, and similar experiments to this one could be applied to other neuron types and other species to see if this is a widespread phenomenon in neuron morphology.

DATA AVAILABILITY STATEMENT

The datasets analyzed for this study can be found in the Open Neurodata AWS account (https://registry.opendata.aws/open-neurodata/). Our package, brainlit provides examples of accessing this data. Specifically, instructions on how to reproduce the figures found here can be found at http://brainlit.neurodata.io/link_stubs/axon_geometry_readme_link.html.

AUTHOR CONTRIBUTIONS

MM and DT advised on the theoretical direction of the manuscript. UM advised on the data application experiments. JV advised on the presentation of the results. TA and JT designed the study, implemented the software, and managed the manuscript text/figures. All authors contributed to manuscript revision.

FUNDING

This work is supported by the National Institutes of Health grants RF1MH121539, P41EB015909, R01NS086888, U19AG033655, the National Science Foundation Grant 2014862, and the National Institute of General Medical Sciences Grant T32GM119998.

ACKNOWLEDGMENTS

We thank the MouseLight team at HHMI Janelia for providing us with access to this data, and answering our questions about it.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fninf.2021.704627/full#supplementary-material

Supplementary Figure 1 | The above plots show the relationship between segment length, and mean curvature or torsion in each segment class and brain. Each data point represents a single axon segment, and average curvature and torsion was computed by sampling the segments at a uniform spacing of 1 μm. We removed segments with zero average curvature/torsion in order to plot the data on a log scale. In this data, there appear to be weak negative correlations between segment length and curvature, and a weak positive correlations between segment length torsion.

REFERENCES

Antonini, A., and Stryker, M. P. (1993). Rapid remodeling of axonal arbors in the visual cortex. *Science* 260, 1819–1821. doi: 10.1126/science.8511592

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

Dierckx, P. (1982). Algorithms for smoothing data with periodic and parametric splines. *Comput. Graph. Image Proc.* 20, 171–184. doi: 10.1016/0146-664X(82)90043-0

Duncan, A., Klassen, E., and Srivastava, A. (2018). Statistical shape analysis of simplified neuronal trees. *Ann. Appl. Stat.* 12, 1385–1421. doi: 10.1214/17-AOAS1107

Grenander, U., Miller, M. I., and Miller, M. (2007). *Pattern Theory: From Representation to Inference*. New York, NY: Oxford University Press.

Heumann, H., and Wittum, G. (2009). The tree-edit-distance, a measure for quantifying neuronal morphology. *Neuroinformatics* 7, 179–190. doi: 10.1007/s12021-009-9051-4

Kunoth, A., Lyche, T., Sangalli, G., and Serra-Capizzano, S. (2018). *Splines and PDEs: From Approximation Theory to Numerical Linear Algebra*. Cham: Springer.

MacLeod, D., Dowman, J., Hammond, R., Leete, T., Inoue, K., and Abeliovich, A. (2006). The familial parkinsonism gene lrrk2 regulates neurite process morphology. *Neuron* 52, 587–593. doi: 10.1016/j.neuron.2006.10.008

Neuhauser, M. (2011). *Nonparametric Statistical Tests: A Computational Approach*. New York, NY: CRC Press.

Parekh, R., and Ascoli, G. A. (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

Portera-Cailliau, C., Weimer, R. M., De Paola, V., Caroni, P., and Svoboda, K. (2005). Diverse modes of axon elaboration in the developing neocortex. *PLoS Biol*. 3:e272. doi: 10.1371/journal.pbio.0030272

Rockland, K. S. (2013). Collateral branching of long-distance cortical projections in monkey. *J. Compar. Neurol*. 521, 4112–4123. doi: 10.1002/cne.23414

Scott, D. W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York, NY: John Wiley & Sons.

Stepanyants, A., Tamás, G., and Chklovskii, D. B. (2004). Class-specific features of neuronal wiring. *Neuron* 43, 251–259. doi: 10.1016/j.neuron.2004.06.013

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* 17, 261–272. doi: 10.1038/s41592-019-0686-2

Winnubst, J., Bas, E., Ferreira, T. A., Wu, Z., Economo, M. N., Edson, P., et al. (2019). Reconstruction of 1,000 projection neurons reveals new cell types and organization of long-range connectivity in the mouse brain. *Cell* 179, 268–281. doi: 10.1016/j.cell.2019.07.042

Conflict of Interest: MM own Anatomy Works with the arrangement being managed by Johns Hopkins University in accordance with its conflict of interest policies.

The remaining 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 26 November 2021
doi: 10.3389/fnins.2021.742652

Creating Detailed Metadata for an R Shiny Analysis of Rodent Behavior Sequence Data Detected Along One Light-Dark Cycle

Julien Colomb[1] *and York Winter*[1,2]*

[1] Department of Biology, Humboldt Universität zu Berlin, Berlin, Germany, [2] Exzellenzcluster NeuroCure, Charité, Berlin, Germany

Automated mouse phenotyping through the high-throughput analysis of home cage behavior has brought hope of a more effective and efficient method for testing rodent models of diseases. Advanced video analysis software is able to derive behavioral sequence data sets from multiple-day recordings. However, no dedicated mechanisms exist for sharing or analyzing these types of data. In this article, we present a free, open-source software actionable through a web browser (an R Shiny application), which performs an analysis of home cage behavioral sequence data, which is designed to spot differences in circadian activity while preventing p-hacking. The software aligns time-series data to the light/dark cycle, and then uses different time windows to produce up to 162 behavior variables per animal. A principal component analysis strategy detected differences between groups. The behavior activity is represented graphically for further explorative analysis. A machine-learning approach was implemented, but it proved ineffective at separating the experimental groups. The software requires spreadsheets that provide information about the experiment (i.e., metadata), thus promoting a data management strategy that leads to FAIR data production. This encourages the publication of some metadata even when the data are kept private. We tested our software by comparing the behavior of female mice in videos recorded twice at 3 and 7 months in a home cage monitoring system. This study demonstrated that combining data management with data analysis leads to a more efficient and effective research process.

Keywords: home cage scan, mus musculus, rodent, automatic, machine learning, multidimensional analysis

OPEN ACCESS

Edited by:
Dezhe Z. Jin,
The Pennsylvania State University (PSU), United States

Reviewed by:
Tudor Constantin Badea,
Transilvania University of Braşov, Romania
Brent Winslow,
Design Interactive, United States

***Correspondence:**
York Winter
york.winter@charite.de

Specialty section:
This article was submitted to Neural Technology, a section of the journal Frontiers in Neuroscience

Received: *16 July 2021*
Accepted: *28 October 2021*
Published: *26 November 2021*

Citation:
Colomb J and Winter Y (2021) Creating Detailed Metadata for an R Shiny Analysis of Rodent Behavior Sequence Data Detected Along One Light-Dark Cycle. Front. Neurosci. 15:742652. doi: 10.3389/fnins.2021.742652

INTRODUCTION

Any attempt to identify the behavioral phenotype of an animal can be a highly tedious undertaking. Animal behavior depends heavily on many variables, which are sometimes uncontrollable, such as general health, age, animal care, sex, environmental factors (pre- and post-natal), housing conditions, environmental stress (including from the experimenter), and diet (Van Meer and Raber, 2005). Therefore, the research community has been searching for high-throughput technologies and methods that can not only phenotype numerous animals through computer automation and with low effort from the experimenter, but also be applied without the experimenter interacting

with the animal. Behavioral analysis of video-captured home cage behavior could potentially be an effective and efficient method for characterizing rodent models of diseases. Because analyzing the behavior of animals under crowded conditions in group housing remains difficult (see Bains et al., 2018 for a review), the most widely used approach is to record animals' behavior in individual cages.

Currently, various technical solutions can provide a detailed analysis of single-housed mouse behavior sequences by analyzing a video (**Figure 1**). These include proprietary systems such as the HomeCageScan (HCS) software (Cleversys, Steele et al., 2007), and the phenorack system (Viewpoint S.A., France Bains et al., 2018), as well as an open source solution (Jhuang et al., 2010), and manual video annotation (see Jhuang et al., 2010 for an example). These software solutions assign one behavior to each video frame (using a short video sequence as the input). The primary data output is a sequence of behavior states. The number of different behaviors recognized varies between the solutions. To simplify the analysis or compare software accuracy, the number of behavior categories can be reduced (Steele et al., 2007; Jhuang et al., 2010; Luby et al., 2012; Adamah-Biassi et al., 2013). In addition to the raw behavior sequence data, the HCS software, as one example, may create summaries of the time spent performing each recognized behavior, as well as of the distance traveled (horizontally) for time intervals from minutes to several days (**Figure 1**).

While much effort has been spent on developing the software that automatically tracks behavioral motives over time, very little effort has been invested into the analysis of the data produced. Published accounts have mostly reported analyses conducted after data were pooled into only two categories and one time window, and were mostly performed manually in Excel (approximately 24 h in Steele et al., 2007, 2.5h before feeding time in Luby et al., 2012). Consequently, such analyses would detect a difference in overall activity, but not in activity types or rhythm that might be relevant (Tobler et al., 1996). The analysis of daily rhythm indeed requires a more careful analysis, making sure the data is synchronized to the daily schedule (light condition changes). On the other hand, the detailed analysis of each behavior leads to a very high number of variables, which require either a multivariate analysis to avoid p-hacking, harking and false positives, or a preliminary experiment to identify the variables of interest *a priori* (Damrau et al., 2021).

Multidimensional approaches have been used previously to separate experimental groups. Steele et al. (2007) used a two-out validation strategy with an L1-regularized logistic regression; specifically, they trained a model on half of the data and then used the model to predict the grouping in the remaining data. This allowed them to discriminate between sick and healthy individuals from the video data well before the appearance of traditionally used symptoms. Another study (Bains et al., 2018) performed a canonical discriminant analysis to select the behavior variables that best separated groups (animal behaviors were monitored manually).

Currently, no repository exists for home cage monitoring data of animal models of disease. For this study, we obtained only

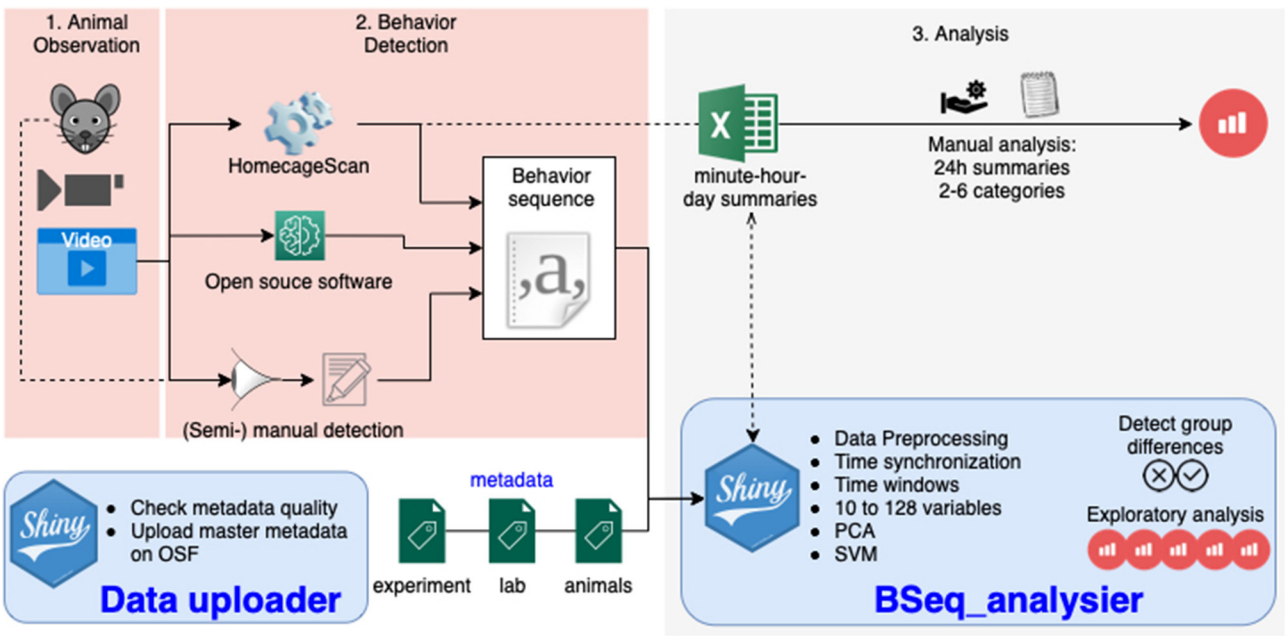


FIGURE 1 | Overview of the workflow in animal homecage behavior analysis: the animal behavior is observed directly or a video is recorded (1), the behavior sequence data is produced manually or by video analysis software (2), and the data is analyzed (3). The tools presented in this paper take care of this third step with one application dedicated to the quality assurance of metadata (providing information about the experiment), and one application analyzing the data to detect differences between groups.

derived data (hourly-binned data exports Steele et al., 2007), because the raw data had not been saved. This restricted our use of meta- and comparative analyses.

In this article, we present an integrated solution for the analysis and management of home cage video monitoring data. We propose a simple metadata schema in the form of spreadsheets that allow for a flexible structure of the data. The data become computer-readable, a first and critical step toward the production of FAIR (findable, accessible, interoperable, and reusable) open data (Group, 2014). In addition, we provide a pack of open-source R scripts and R Shiny applications (apps) that can analyze such FAIR data. On top of being available for use and further development, the BSeq_analyser application (**Figure 1**) is provided with an easy to use interface. It accounts for both daily rhythms (synchronizing data along the day/night cycle of the animal and splitting it into time windows of identical size for each animal) and activity spectrum (with a minimal pooling of behavior categories), producing up to 162 variables per experiment. It runs a multidimensional analysis that tests whether different experimental groups can be distinguished (using the first component after a principal component analysis [PCA] or based on a machine-learning strategy). It also provides plots of hourly activities for explorative analysis.

We tested the software using unpublished data obtained in Berlin, as well as previously published data obtained from Andrew Steele's lab. In particular, we compared the behavior profile of animals monitored twice at 3 and 7 months of age. Because of the differences in age and experience, we expected a change in behavior, which our analysis was in fact able to detect.

MATERIALS AND METHODS

Data Provenance and Animal Testing

The authors did not perform the animal research described in the manuscript but only analyzed the data. The data used in this manuscript was collected by the animal outcome core facility in Berlin and Prof. Steele's group, as described in the master metadata file, following the method described in Schroeder et al. (2021) and Steele et al. (2007), respectively. In brief, the natural behavior of single mice within a home cage, unaffected by an experimenter, was video- recorded from a side view. Animals were singly housed for approximately 23 h in a regular home cage (EU type II) without additional enrichment (to avoid the detection of artifacts on video), but with free access to food and water. The videos were analyzed to classify the single behavior shown on each frame using the HCS software package (CleverSys Inc., USA).

Software and Data Availability

We used Rstudio and GitHub to develop the open source software (MIT licensed) as well as to organize its development and version control (www.github.com/jcolomb/HCS_analysis). Github issues were used to archive some discussions held with the CleverSys staff and data providers (Andrew Steele). Different milestones of the development were and will be archived on Zenodo to assure long-term preservation of the software (doi: 10.5281/zenodo.1162721). Data were added to the repository. Different text files available with the software describe and document the use of the two apps, details of the analysis algorithms, and ways to expand the analysis. A readme file explains how to navigate them.

Main Dependencies

The software was built on R resources (R Core Team, 2020). This work would not have been possible without the tidyverse environment (Wickham, 2019), packages for interactive processing (Chang et al., 2020; Pedersen et al., 2020; Sievert et al., 2020), statistical analysis (Breiman et al., 2018; Helwig, 2018; Park and Hastie, 2018; Meyer et al., 2019; Harrell, 2020) and graphical interfaces (Auguie, 2017; Murrell, 2020; Sievert et al., 2020). It also depended on the osfr package, which was still in development (Wolen and Hartgerink, 2020) and loaded via the devtools package (Wickham et al., 2020). We used the env package (Ushey, 2020) to dock the project.

Metadata Structure

The metadata was structured in different files to avoid having to provide the same information multiple times (**Figure 2**). Each experiment was described in the master metadata file available online (https://osf.io/myxcv/). We expanded the RADAR descriptive metadata schema (Kurze et al., 2017) to create the structure of the master metadata (**Table 1**). The information entered in that file was made openly available even if the data was not. In addition to the generic entries from RADAR, the file contained information about the path to the other three metadata files - the experiment (one row per test provides details about the animal and the experiment), lab (conditions such as light conditions are given), and identifiers metadata file - and the data folder, as well as information about the software used to acquire and analyze the video.

Metadata and Data Registration

We have provided a detailed manual that describes the relatively complex metadata creation process (see the readme file), and also a Shiny app for testing the quality of the metadata entered by users (available at analysis/Shiny__testanduploaddata/) and pushed it to OSF. We followed the manual and push metadata from a different experiment performed in Berlin.

The master metadata file was deposited on the open science framework storage at "https://osf.io/myxcv/" via the Shiny app. We chose this solution not only because we could read and update it directly from R, but also because it was version controlled (i.e., misuse will not have serious consequences). This file indexed all experiments that were analyzed with the software, but the deposition of the actual data remained independent and optional. The analysis software could access data locally (as in the example provided) or on the web via the HTTP protocol. We used the Github repository as one practical example.

Data Analysis

The detailed process of the analysis can be read directly from the commented code and readme file available on Zenodo and GitHub. Variables can be entered in the master_noshiny.r file or via the Shiny app, and the master_shiny.r file is then processed.

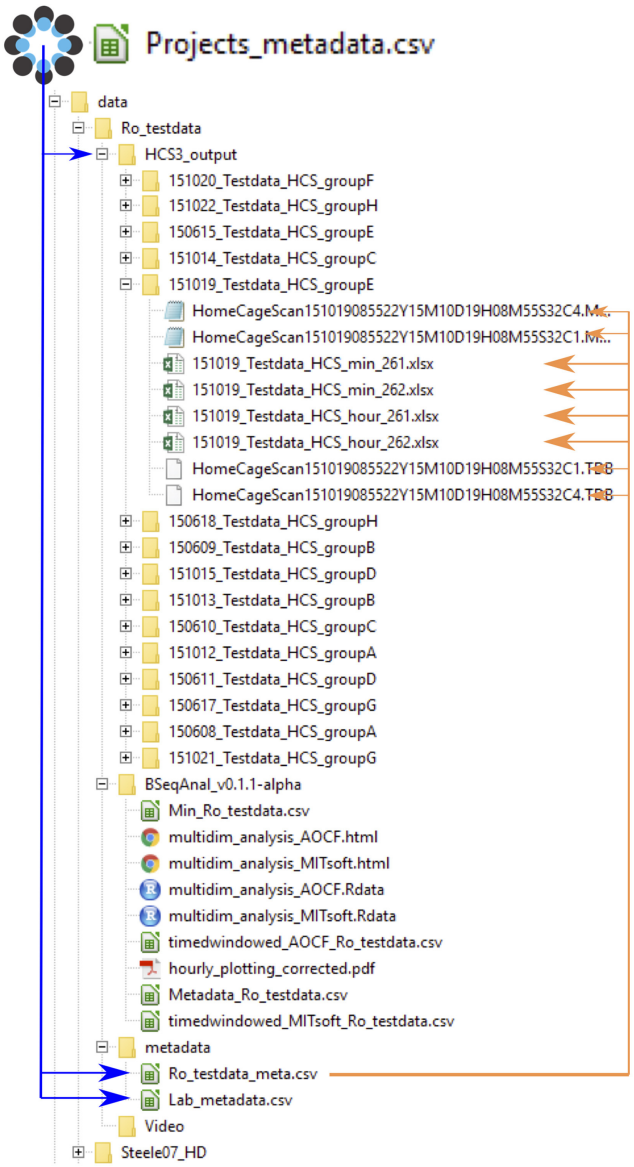


FIGURE 2 | Data and metadata structure. The master project metadata file links the address of the metadata files and the data folder. The experiment metadata file links to each data file (for clarity, only one folder is shown here). The format of the data is either .xlsx summary files (min or hour) or the HCS output files .mbr (behavior sequence) and .tbd (position; note that the software does not read that file). By reading the master file, the computer can determine the path to every data file. Upon analysis, the software creates a new folder indicating the software's name and version. Its reports are saved there, while derived data files are saved in a folder named after the software name but not its version.

TABLE 1 | Master metadata information.

Identifier	**F0001**
Proj_name	Ro_testdata
Title	Wild type data at different age. For testing purpose
Creator	Colomb, Julien
Contributors	Long, Melissa; Winter, York (https://orcid.org/0000-0002-7828-1872)
Creator_email	julien.colomb@fu-berlin.de
Publisher	
Publication year	
Production year	2015
Subject area	Behavioral neurobiology
Resource	Dataset
Rights	CC0
Rights holder	Winter, York
Description_ comments	Part of a project at the AOCF, only data from wild type animals are available here.
Funder information	XXX
video_acquisition	HCS 3.0
video_analysis	HCS 3.0
group_by	Treatment
confound_by	
source_data	this_github
Folder_path	Ro_testdata
raw_data_folder	HCS3_output
video_folder	Videos
animal_metadata	metadata/Ro_testdata_meta.csv
lab_metadata	metadata/Lab_metadata.csv
indentificator_ metadata	

The second tab in the Shiny app plots hourly summary data by running the "plot5 hoursummaries.r" code. A brief description of the software procedure is provided below.

Overview

The analysis software automatically reads the master metadata file on OSF. When the user specifies the project to be analyzed, the software will import, process and analyze the data (**Figure 3**). The software reads the metadata associated with the project and creates a synchronized minute summary file from the indicated primary data file (raw data or minute/hourly summary files). The minute summary is a table where each row reports the amount of time spent performing each behavior for each minute of experiment, the time relative to the start of the experiment, the time relative to the light extinction, and the animal ID and group. Behavior categories (see **Table 2**) are pooled and the software creates time windows (only time windows where all animals have data can be produced, the user can choose which time window to incorporate in the analysis), before calculating a value for each behavior category for each time window. Some data might be excluded from the analysis at this point, following the label indicated in the experiment metadata. The software then performs multidimensional analyses on this window's summary data to plot them as well as to test whether experimental groups can be distinguished. The analysis involves a random forest (RF) analysis for identifying the variables that exhibit the largest differences among the different groups of mice. Next, an independent component analysis (ICA) is performed on these 8–20 variables and the first three components are plotted in an interactive 3D plot. Independently, the next part of the software

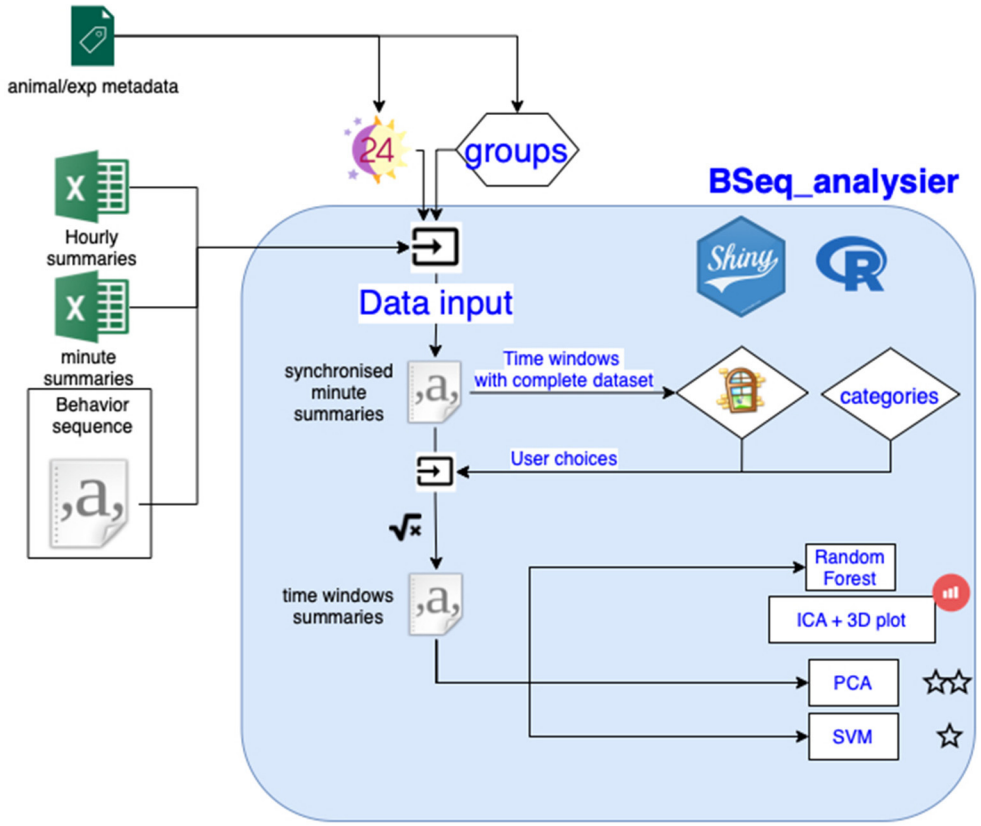


FIGURE 3 | Overview of the data analysis workflow. See text for details.

runs a PCA and examines the first principal component for statistically different results in the groups using a nonparametric test. Then, it may run a machine-learning algorithm on the data using a support vector machine (SVM) algorithm. Validation of the latter results is conducted through a non-exhaustive two-out validation technique as in Steele et al. (2007) if the sample size per group is below 15, or otherwise through a test data set. Hourly summary plots (using the synchronized minutes summary data) are also provided in the application second tab.

Data Inputting Details

The software reads the **minute summary** file created by the HCS software or creates a new one from the raw behavior sequence data or the hourly summary data. In the latter case, the hour value divided by 60 is used for each minute of that hour. The software adds a column that indicates the time to the light-off event ("bintodark") and what the light condition was (DAY or NIGHT). This is calculated from the start of the experiment in the experiment metadata (which can be read from the name of the video file coming from the HCS software package) and the light/dark cycle information obtained from the lab metadata file.

We used the information delivered by CleverSys to derive **categories** from the raw sequence files code, and obtained 38 categories (the distance traveled on the x axis was not considered a behavior category; No Data and Arousal were discarded; and six different drink and eat categories were pooled into two). The synchronized minute summary file is saved on the hard-disk at this point, and will be read by the software on a subsequent run.

In the next step, the software pools these 38 categories into 18 (Berlin categories: we restricted the number of categories to pool some behavior types that are very rarely detected) or 10 (Jhuang categories: categories the Jhuang open source software can detect) using the "grouping variables.r" code (**Table 2**). The data records were typically from experiments lasting slightly less than 24 h. Nine different **time windows** were defined, with the last three windows overlapping with the first six; see **Figure 4**.

Then, the square root of the proportion of time spent performing each behavior during each time window is calculated. This **data transformation** makes the data more normally distributed and allows for a better analysis in a multidimensional space, but it does not require non-null values like log transformation. This derived data set is the multivariate dataset (called "Multi_datainput" in the code, time windows summaries in **Figure 3**), which can contain up to 162 variables per subject (18 behavior categories times 9 time windows), it is also saved on disc.

Multivariate Data Visualization and Analysis

The software uses a double RF analysis to select 8–20 variables, which are used as input for ICA. The first RF selects the best 20

TABLE 2 | The initial 45 categories from the HCS outputs were pooled into 18 and 10 categories, the latter being the only categories available in another open source video analysis software, while the former was used to pool categories that have very little occurence.

Original_HCS	Berlin_category	Jhuang_category
Travel.m.	Distance_traveled	Distance_traveled
ComeDown	ComeDown	Rear
RearUp	Rearup	Rear
Turn	Walk	Walk
Stretch	Stretch	Rear
HangCudl	Hang	Hang
HangVert	Hang	Hang
CDfromPR	ComeDown	Rear
CDtoPR	Rearup	Rear
RUfromPR	Rearup	Rear
RUtoPR	Rearup	Rear
LandVert	Hang	Rear
WalkLeft	Walk	Walk
WalkRght	Walk	Walk
Stationa	Immobile	Rest
Drnk.S1.	Drink	Drink
Eat.Z1.	Eat	Eat
Jump	Jump	Unknown_behavior
Unknown	Unknown	Unknown_behavior
HVfromRU	Hang	Hang
HVfromHC	Hang	Hang
ReptJump	Jump	Unknown_behavior
Circle	Walk	Walk
Dig	Digforage	Unknown_behavior
Forage	Digforage	Unknown_behavior
Pause	Immobile	Micro_move
Urinate	Unknown	Unknown_behavior
Groom	Groom	Groom
Sleep	Immobile	Rest
Twitch	Twitch	Micro_move
Arousal		
Awaken	Awaken	Micro_move
Chew	Chew	Eat
Sniff	Sniffing	Micro_move
RemainRU	Rearup	Rear
RemainPR	Rearup	Rear
RemainHV	Hang	Hang
RemainHC	Hang	Hang
RemainLw	RemainLow	Micro_move
WalkSlow	Walk	Walk
No.Data		
Drnk.S2.	Drink	Drink
Drnk.S3.	Drink	Drink
Eat.Z2.	Eat	Eat
Eat.Z3.	Eat	Eat

variables, whereas the second RF is performed using only these 20 variables. The best eight variables or all variables with a Gini score above 0.95 are kept for the ICA and are listed in the report. The data are then plotted according to the first three components of the ICA, resulting in a three-dimensional plot.

Then, the software performs a statistical analysis using a **PCA** on the multivariate data set, and then plots the first component and performs a statistical analysis of this first component over groups. Finally, the user can choose (via the "Perform the multidimensional analysis (takes time)" button) to perform a **machine-learning analysis** based on a SVM approach using a radial kernel. We also attempted an L1-regularized regression, modifying the code used in Steele et al. (2007), obtained from Prof. King. The models were used to predict the experimental group of the data not used for training. The software used two different **validation techniques**. For data sets with fewer than 15 animals per group, a two-out validation strategy is used, whereas the software uses a completely independent test data set when the sample size exceeds 15 (see the analysis_details.md text delivered with the software for details). The software reports the kappa score as a measure of model accuracy. For the statistical analysis, the same machine-learning code is run on the same data but after a randomization of the group (permutation). This provides us with a cloud of accuracy results that can be used to perform a binomial test, which in turn provides us with a p-value that indicates whether the model can predict the experimental group at a level above chance.

RESULTS

Data Integration

In order to facilitate the analysis of data from different sources, we proposed a format for organizing the data (behavior sequence or binned summary data) and the metadata (information about the experiment, the lab, and the animals), such that the R Shiny apps can access the different files automatically. Critically, this format does not require any file to be renamed, but it does include file names in the experiment metadata. An extra main and public metadata file reports information about the project, its contributors, and the placement of the other files (see **Figure 2**), making the data FAIR (Group, 2014).

We have also provided a walk-through (available at metadata/information/Readme.md) and a Shiny app to facilitate new data integration. The app tests and uploads the project metadata to the master metadata file online (which can then be read by the second app devoted to data analysis). The process of creating the spreadsheets lasts for approximately 1 h once all information have been gathered. The data files themselves did not require any modification. We obtained data of different quality and formats from different labs. The software deals only with files output from the HCS software package (CleverSys Inc.) thus far (the raw behavior sequence [.mbr file] or the minute or hour binned data summaries).

We provided and used a data set produced in Berlin of 11 wild-type female mice recorded twice (at the age of 3 and 7 months, respectively) for approximately 24 h, and published data obtained from Andrew Steele (Steele et al., 2007; Luby et al., 2012). Other data sets were tested but the data were not made public. The sample size was decided independent of this study and one animal was excluded because the data for one time

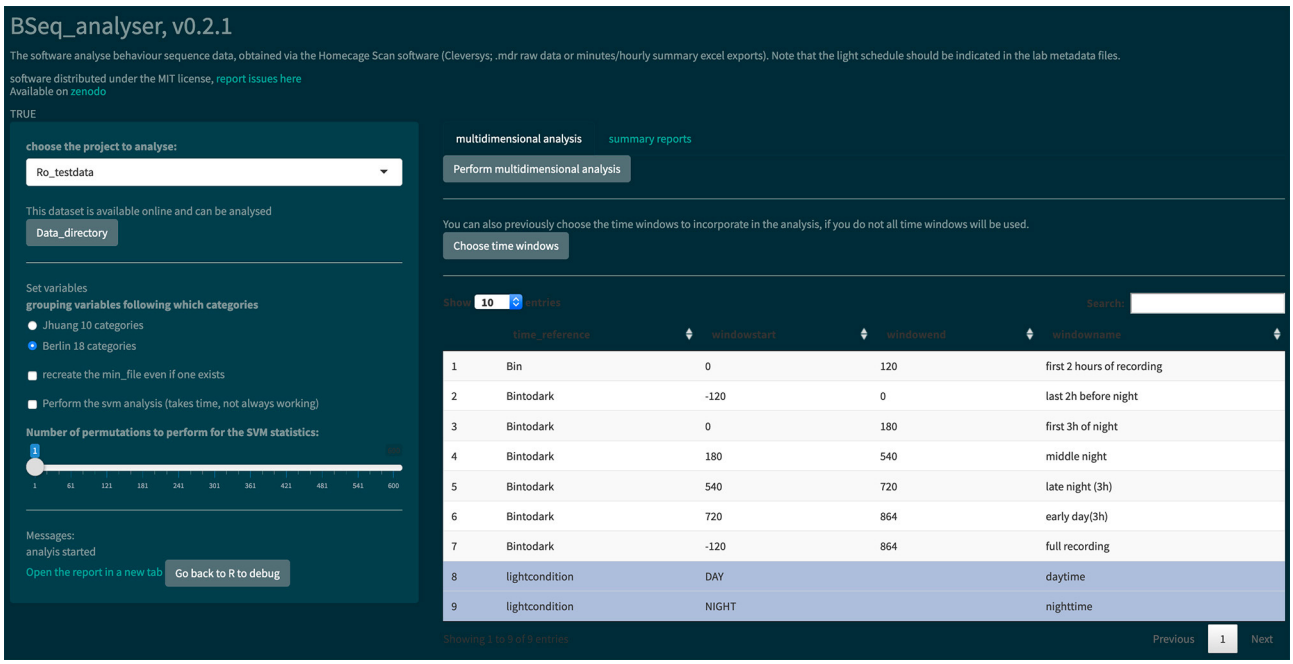


FIGURE 4 | Screenshot of the Bseq_analyser Shiny app for data analysis. In the left panel, the user indicates the data and variables to use. He or she can also indicate where to find the data that are not published online (by clicking the "Data directory" button). The left panel also presents some messages and a link to the report. The main panel has two tabs, one for performing the multidimensional analysis (with a prior choice of time windows or not) and one for creating hourly summary plots of each behavior category.

point were not available. Mice were tested in the same order at the two time points, and were subjected to other behavioral tests in the 4-month time period between the two home cage monitoring events.

Data Analysis

We used data obtained using the HCS software with 11 wild-type mice recorded twice for approximately 24 h. The data were grouped following the age of the animal (young or old) at the time of the recording (available under Ro_testdata project). One Excel export file was corrupted (animal 279, first test), whereas the data of one animal was inconsistent (animal 25, second test: the raw data and the exported data did not correspond). Animal 25 was removed from the analysis by modifying the metadata file, which contained an "exclude" column.

The BSeq_analyser R Shiny app was used to analyze the data, as shown in **Figure 4**. In the left panel, the user must choose variables: the project to analyze, the behavior categorization to use (**Table 2**), whether to recreate the minute summary file from the raw data, whether a machine learning analysis should be performed, and the number of permutations to perform (if machine learning analysis is performed). The users might then choose which time windows to incorporate in the analysis. They then press the "Perform multidimensional analysis" button and wait until the html report is produced and presented on screen. We performed this analysis once with the corrupted data from the Excel summary files (**Figure 5**) and once with a corrected

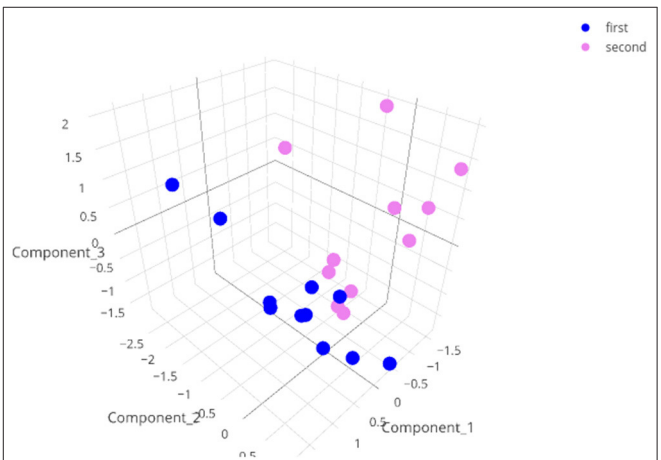


FIGURE 5 | Three dimensional representation of the results of the ICA on the test data with a minute summary as the primary data. Note that the data with corrupted entries (animal 279, first test) does not show up as an outlier in this graph.

export of the raw data (see whole report at https://doi.org/10.6084/m9.figshare.6724547.v3).

The software read and transformed the data according to the information given in the metadata and the variables selected. It reports a data analysis using a PCA, present the results of RF analysis and visualizes the data in 3 dimensions. (for details, see

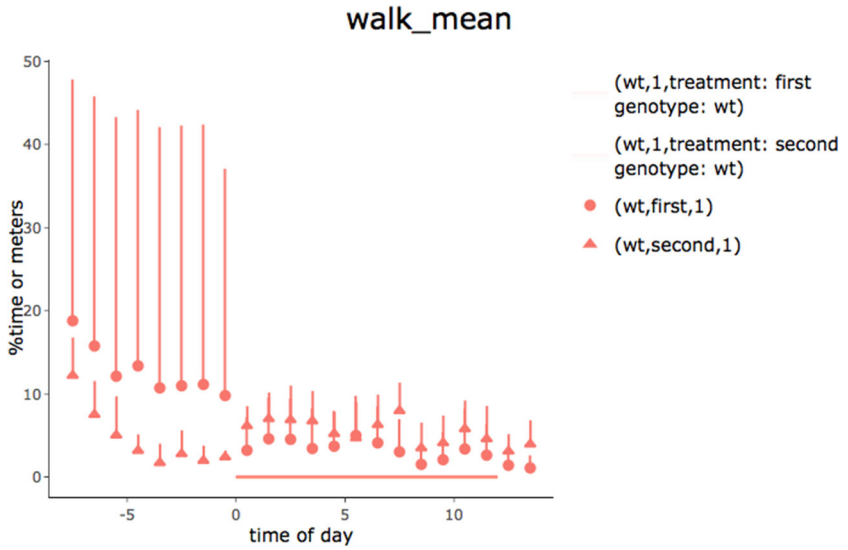


FIGURE 6 | Time spent walking for young (first) and old (second) animals observed for 24 h (lights went off at 0 h). Recordings began immediately after the animals had been placed in the observational cages. Data are means and standard deviation expressed as percentages of time spent walking. The increase in activity at the start of the test appeared to wear off faster in old animals. In order to work for different types of grouping, the legend shows genotype and treatment conditions and is not optimized.

the Materials and Methods section and the code itself). The html report is saved on the hard disk (see **Figure 4**) and can be directly opened in the browser app. Noteworthily, the PCA was effective at separating the two experimental groups (nonparametric statistical test on the first component [$p = 0.00067$; the effect size was large: Z/square(n) = 0.76]).

For the data visualization, a random forest algorithm was used to choose the 8–20 variables that were the most effective at separating the different animal groups. An independent component analysis (ICA) was then run on these variables and the data were plotted in two or three dimensions. When we performed this analysis including the corrupted file, the data point was surprisingly not an extreme value (see **Figure 5**, interactive at https://plot.ly/j_colomb/39/).

In the second tab of the app, hourly summaries of the percentage of time spent performing each behavior are provided (time is synchronized to the light-off event). Although it can be directly seen in the app (plot by plot), a pdf file with all plots is also produced. **Figure 6** presents an example of a 24-h summary plot obtained using the "walking" behavior category.

Machine Learning Analysis

The software predicts group separation using a multidimensional analysis. In addition to PCA, it might then use a supervised machine learning (support vector machine [SVM] using a radial kernel) approach to separate the two groups. Noteworthily, the software can also deal with three different experimental groups, but no more (the data had to be split into pairs of groups and an analysis was performed for each pair). The SVM is trained on part of the data. The model is then used to predict the group membership of data not used to train the model. The kappa score gives an indication of the effectiveness of the model, which itself indicates how easy the two groups of data can be separated. The software uses either a two-out validation strategy (as in Steele et al., 2007) if there are fewer than 15 animals per group, or an independent test data set otherwise. The whole process is repeated after permuting the group membership of the train data set. A binomial test compares the actual accuracy with a cloud of accuracy values obtained after many permutations, thus calculating a range for the p-value. The number of permutations is reported with this estimated p-value.

In order to test the efficacy of the approach, we ran the analysis using different variables with our test data set. This was performed with version v0.1.1-alpha of the software and with the two corrupted files for animals 25 and 279. While the PCA could tell the experimental groups apart (data not shown), the machine-learning approach was not as effective. We performed analyses over three time window variations: one time window (from 2 h before lights off to the end of the recording), five time windows (first 2 h of recording, last 2 h before nightfall, first 3 h of the night, last 3 h of the night, first 3 h of the second day), or six time windows (all of the windows described above). We ran the analysis using both behavior categorizations. Since the number of animals was low (11 per group), we used the two-out validation procedure. The algorithm could tell the two groups apart when the Berlin categorization was used, but not when the Jhuang categorization was used, irrespective of the time window combination or algorithm used (**Figure 7**). When the same analysis was performed with corrected data, the latest code and the one time window in the Berlin categorization seemed to provide an even worse success rate for the SVM approach https://doi.org/10.6084/m9.figshare.6724547.v3.

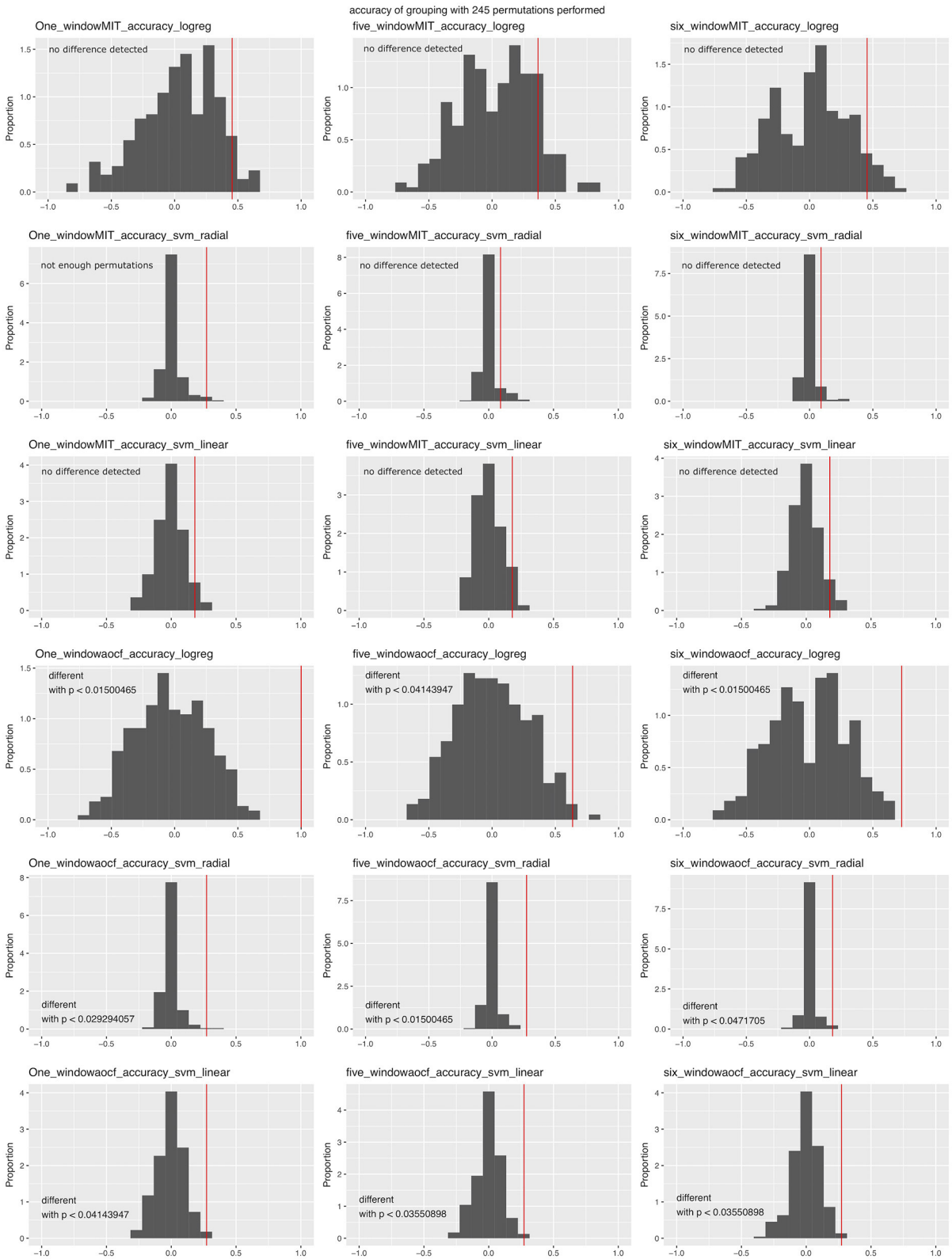


FIGURE 7 | Accuracy of machine-learning algorithms in predicting data group membership in the test data, using two-out validation. The red line represents the accuracy when the real groups were used, whereas the distribution represents the accuracy obtained when data group membership was randomized prior to the

(Continued)

FIGURE 7 | analysis. Graphs are grouped according to the number of time windows used in the columns (one, five, or all six windows; see text), categorization of the behavior (first three rows: Jhuang categorization, last three rows: Berlin categorization) and the machine-learning algorithm (L1-regularized regression: rows 1 and 4; SVM with radial kernel: rows 2 and 5; SVM with linear kernel: rows 3 and 6); p-values were obtained through confidence intervals for binomial probability analysis.

Working With Hourly Summaries and Raw Data

The software could also use hourly summaries as primary input data (they are the only data available in the Steeleo7 HD data set). In this case, a minute summary was produced by dividing the hourly value by 60. The synchronization with lights off between experiments was not precise in those cases, but a rough analysis of the output revealed that this had only a minor effect on the whole analysis (https://doi.org/10.6084/m9.figshare.6724604.v1).

In general, we recommend exporting minute summaries from the HCS software for new experiments (to obtain the distance traveled) but using the raw data for analysis. Indeed, the distance traveled per minute cannot be calculated with our software. However, the created minute summary file is more robust than that from the HCS software; specifically, some behavior events were sometimes not taken into account, and in one case the HCS export function failed completely.

Remarkably, using the raw data as the input allows for a more complex analysis of the data. One can, for instance, analyze the transition between different behaviors. For example, we showed which behavior was performed before and after "land vertical" events, merging our two experimental groups (**Figure 8**). While landing occurred after hanging behaviors as expected, the animals started to either rear again or engage in sniffing or eating behaviors, but rarely started to walk directly after a landing. In addition, the "hanging vertical from the rear up" behavior notably did not follow a rear up behavior in these cases.

Meta-analysis

In order to test the re-usability of our data and code, we performed a meta-analysis using data from different projects. We read all data at our disposal for wild-type animals. We then performed the usual analysis with all nine time windows, followed by a PCA (we could not include the Steele07_HD data because the birth date of the animals was not provided, and also because the seventh time window did not have data). We plotted the first PCA component against the age of the animal, adding the genotype as an additional variable (**Figure 9**). The results suggested that both age and genotype might affect mouse behavior in the home cage.

DISCUSSION

FAIR Data per Default

Using relatively simple tools (R and spreadsheets) and common platforms (GitHub, OSF, and Zenodo), we combined data analysis and data "FAIRification" into one workflow. On top of metadata necessary for the data analysis, we ask the users to provide general information about the experiment, and strongly encourage them to publish this particular piece of metadata through one of our apps (**Figure 2**). This creates an open repository for home cage monitoring metadata in a spreadsheet form (https://osf.io/myxcv/). Users may choose to keep the data private, but even unpublished data is in a state to be shared easily.

Home cage monitoring experiments lead to videos that are analyzed to produce a time stamped sequence of recognized behaviors. By combining these data with metadata (which provide information about the experiments and the experimenters in a computer-readable form), one can produce interesting visualizations and analyses, especially if the raw data (in this case an .mbr text file) are provided. We encourage users to avoid using the Excel summary files produced by the proprietary software, but rather to start the analysis from the raw data. Doing so will make the analysis more robust: data from different software may be included more easily, and one avoids problems created in closed access export functions. In particular, we encourage users not to include the distance traveled variable in the analysis, as its spread differs from the other variables (percentage of time spent performing a behavior) and thus including it in a multidimensional analysis may cause problems.

In order to best test the data and code re-usability, we performed a meta-analysis (**Figure 9**). We pooled all data available to us, filtered those from wild-type strains, and asked whether animal age or genotype had the most influence on animal behavior. While the amount of data available to us proved insufficient to answer the question, the analysis could be performed with few issues.

Data Visualization and Analysis

The software aligns the data to the light/dark cycle, and then cleans the data to only keep data points where all subjects have provided data, thus ensuring that each sample is of equivalent valence for the analysis. Such data cleansing has been absent from most analyses published to date, although it might be crucial for spotting specific effects at the time of the light/dark switch. We also implemented different time windows to create specific variables along the day/night cycle, in order to detect differences that could be overseen with a 24-h summary analysis. We are confident that the software represents progress toward a cleaner and more detailed analysis of behavior sequence data.

In addition, the use of the software would prevent p-hacking and harking (Kerr, 1998). To illustrate this, we shuffled the grouping of the test data before performing the analysis (data not shown). We observed that the 3D data visualization still showed some differences between the groups. This was expected because the RF analysis is meant to look for the cause of differences and would find some in a data set with 162 variables. The summary analysis (which corresponds to the type of analysis usually performed) can still reveal some apparent differences for some behavioral traits – differences that could be claimed to be statistically significant if one does not correct for multiple testing.

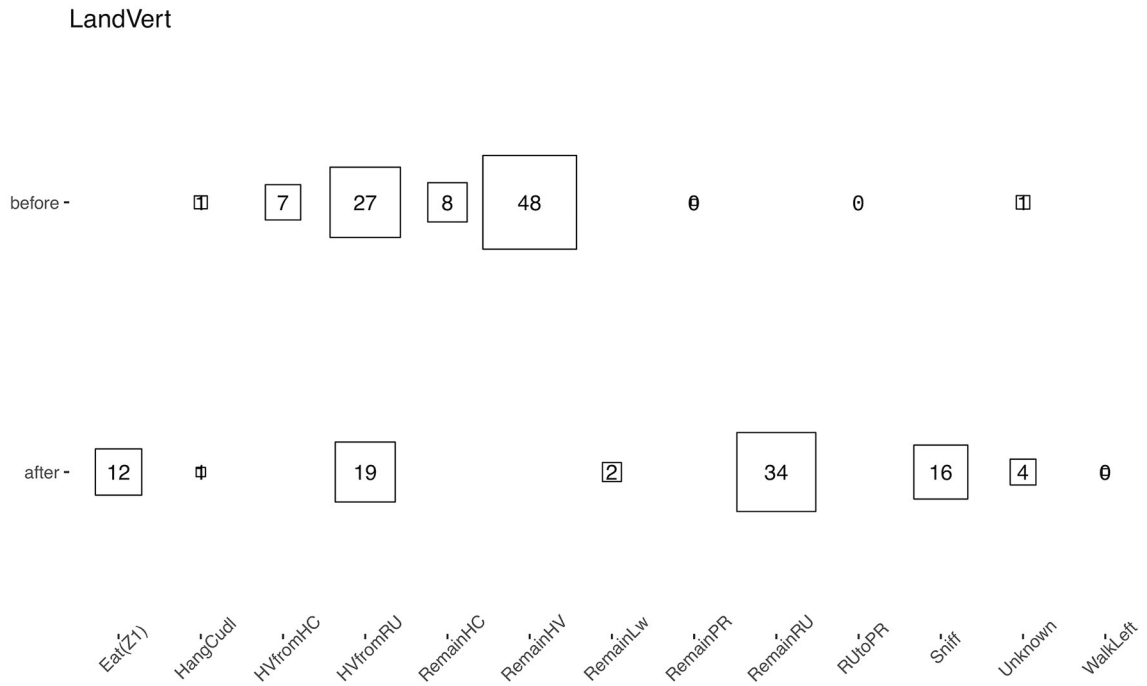


FIGURE 8 | Average percentage of time that a behavior (the original HCS behavior categories were used) appeared just before or after a "land vertical" behavior. The eight behaviors with the highest median proportion are shown; squares and numbers represent the mean percentage. Similar numbers were obtained when taking the median.

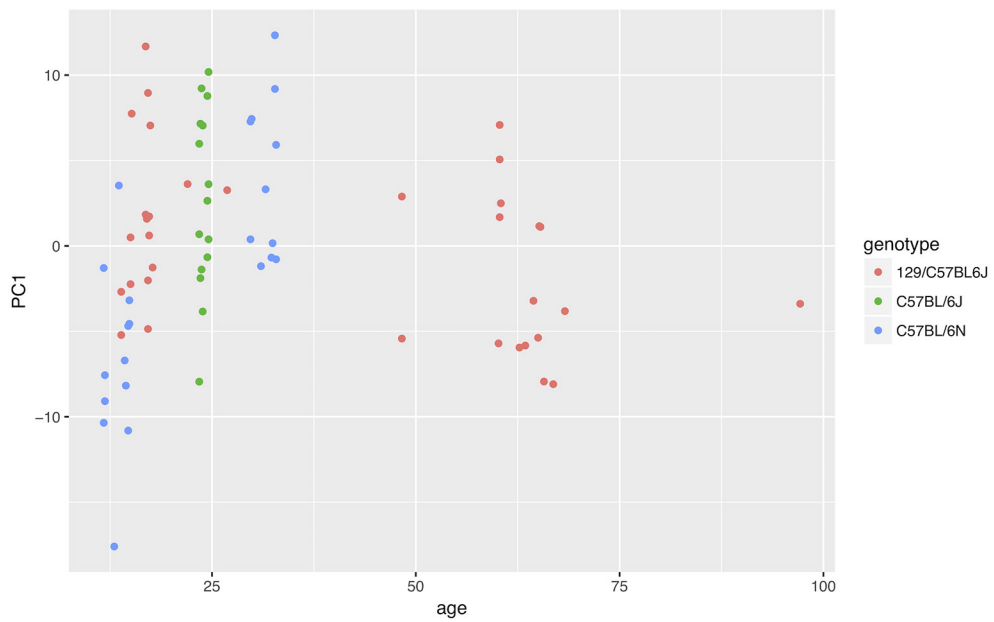


FIGURE 9 | Effect of age and genotype on behavior phenotype as summarized by the first component of a PCA. The data suggested that both genotype and age could affect animal behavior.

However, the PCA clearly indicated that a difference between the two artificial groups of data could not be detected statistically, as expected.

We made quite an effort to implement and test a machine-learning approach, with the idea being that a PCA may miss existing differences due to high variability for some variables

inside groups. However, our analysis revealed that this approach seems to be ineffective with our type of data (**Figure 7**). In particular, the analysis revealed that the distribution of the accuracy of predictions in randomly permuted groups varied greatly between algorithms, which questions the approach used by Steele et al. (2007).

An Open Source Proof of Concept

By using a GitHub workflow and an open-source programming language (R), providing Shiny apps for use by non-coders, and implementing metadata in simple spreadsheets that are easy to read and write, we hope to reach the growing community of researchers who are dealing with behavioral sequence data. The software is intended for non-computer-scientist researchers to read and extend, and therefore, it has been kept simple. While we have provided extensive comments, including dependencies, as well as a hierarchy of code files to facilitate code reading, we did not use functions nor implement tests. Similarly, the experiment metadata are provided in spreadsheets, a practical solution that we were able to implement with little effort. We believe that the implementation of a more complex data format would be counterproductive at this stage.

The analysis runs identically on the Shiny app or when variables are provided in a code file, so debugging and extension creation can be performed without the need to care about the difficulties of Shiny apps debugging. We used that approach to perform a quick analysis of the behavior transition in our data set. Our results demonstrated the potential of this approach both for spotting limits in the video analysis software (e.g., inconsistent sequences) and for creating new, more detailed analyses based on the behavior sequence itself.

Mouse Behavior

As expected, mouse behavior differed in the second session compared with the first session, which was detected by a PCA. An explorative look at the data suggested that mice are more active at the beginning of their first session, during the day, confirming that the use of different time windows is beneficial for the analysis of the data. Our meta-analysis also suggested that both age and genotype influence mouse behavior.

CONCLUSION

We have presented several open-source Shiny apps that allow the archiving, visualization, and analysis of long-term home cage video monitoring experiments. This report is a proof of concept for workflows allowing both data analysis and publication. The analysis tool by itself should be helpful for the analysis of behavioral sequence data. It cleanses the data before analysis and provides an easy way to test for group effects including patterns in circadian behavior, while avoiding harking and p-hacking. We hope that the community will increase the amount of data openly available as well as expand the software in novel ways for analyzing behavioral sequence data.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

JC: conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, validation, visualization, writing–original draft, writing–review, and editing. YW: conceptualization, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, visualization, and writing–review, and editing. Both authors contributed to the article and approved the submitted version.

FUNDING

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), SFB 1315, Project-ID 327654276, and EXC 2049: NeuroCure, Project-ID 390688087. We acknowledge support from the Open Access Publication Fund of Charité – Universitätsmedizin Berlin.

ACKNOWLEDGMENTS

We wish to thank the members of the lab and the AOCF team: Andrei Istudor for his suggestion to integrate all used variables in the software report; Patrick Beye for the discussion on the design of the analysis; Melissa Long for performing the home cage monitoring experiment and for helping with the creation of the Metadata; and Vladislav Nachev for the scientific input and discussion. In addition, we wish to thank Prof. Steele and Prof. King for their fruitful discussions and access to their code and data, and also CleverSys Inc. for their help with decoding the HCS outputs.

REFERENCES

Adamah-Biassi, E., Stepien, I., Hudson, R., and Dubocovich, M. (2013). Automated video analysis system reveals distinct diurnal behaviors in C57BL/6 and C3H/HeN mice. *Behav. Brain Res.* 243, 306–312. doi: 10.1016/j.bbr.2013.01.003

Auguie, B. (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. R package version 2.3.

Bains, R. S., Wells, S., Sillito, R. R., Armstrong, J. D., Cater, H. L., Banks, G., et al. (2018). Assessing mouse behaviour throughout the light/dark cycle using automated in-cage analysis tools. *J. Neurosci. Methods* 300, 37–47. doi: 10.1016/j.jneumeth.2017.04.014

Breiman, L., Cutler, A., Liaw, A., and Wiener, M. (2018). *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. R package version 4.6-14.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2020). *shiny: Web Application Framework for R*. R package version 1.5.0.

Damrau, C., Colomb, J., and Brembs, B. (2021). Sensitivity to expression levels underlies differential dominance of a putative null allele of the

Drosophila tβh gene in behavioral phenotypes. *PLoS Biol*. 19:e3001228. doi: 10.1371/journal.pbio.3001228

Group, D. C. S. (2014). *Data Citation Synthesis Group: Joint Declaration of Data Citation Principles*. Technical report.

Harrell, Jr., F. E. (2020). *Hmisc: Harrell Miscellaneous*. R package version 4.4-0.

Helwig, N. E. (2018). *ica: Independent Component Analysis*. R package version 1.0-2.

Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A. D., et al. (2010). Automated home-cage behavioural phenotyping of mice. *Nat. Commun*. 1, 1–9. doi: 10.1038/ncomms1064

Kerr, N. L. (1998). HARKing: hypothesizing after the results are known. *Pers. Soc. Psychol. Rev*. 2, 196–217. doi: 10.1207/s15327957pspr0203_4

Kurze, T., Tobias, R., and Bonn, M. (2017). "One-stop publishing and archiving: forschungsdaten für Promotionsvorhaben über Repositorien publizieren und archivieren: eine landesweite Initiative im Rahmen des Projekts bwDataDiss am Beispiel des Karlsruher Instituts für Technologie (KIT)," in *E-Science-Tage 2017: Forschungsdaten Managen. Hrsg*., ed J. Kratzke (heiBOOKS). Available online at: https://publikationen.bibliothek.kit.edu/1000078226

Luby, M. D., Hsu, C. T., Shuster, S. A., Gallardo, C. M., Mistlberger, R. E., King, O. D., et al. (2012). Food anticipatory activity behavior of mice across a wide range of circadian and non-circadian intervals. *PLoS ONE* 7:e37992. doi: 10.1371/journal.pone.0037992

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2019). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-3.

Murrell, P. (2020). *RGraphics: Data and Functions from the Book R Graphics, Third Edition*. R package version 3.0-2.

Park, M. Y. and Hastie, T. (2018). *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*. R package version 0.98.

Pedersen, T. L., Nijs, V., Schaffner, T., and Nantz, E. (2020). *shinyFiles: A Server-Side File System Viewer for Shiny*. R package version 0.8.0.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.

Schroeder, P., Rivalan, M., Zaqout, S., Krüger, C., Schüler, J., Long, M., et al. (2021). Abnormal brain structure and behavior in MyD88-deficient mice. *Brain Behav. Immun*. 91, 181–193. doi: 10.1016/j.bbi.2020.09.024

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., and Despouy, P. (2020). *plotly: Create Interactive Web Graphics via 'plotly.js'*. R package version 4.9.2.1.

Steele, A. D., Jackson, W. S., King, O. D., and Lindquist, S. (2007). The power of automated high-resolution behavior analysis revealed by its application to mouse models of Huntington's and prion diseases. *Proc. Natl. Acad. Sci. U.S.A*. 104, 1983–1988. doi: 10.1073/pnas.0610779104

Tobler, I., Gaus, S. E., Deboer, T., Achermann, P., Fischer, M., Rülicke, T., et al. (1996). Altered circadian activity rhythms and sleep in mice devoid of prion protein. *Nature* 380, 639–642. doi: 10.1038/380639a0

Ushey, K. (2020). *renv: Project Environments*. R package version 0.11.0. Available online at: https://CRAN.R-project.org/package=renv

Van Meer, P., and Raber, J. (2005). Mouse behavioural analysis in systems biology. *Biochem. J*. 389(Pt 3):593–610. doi: 10.1042/BJ20042023

Wickham, H. (2019). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.3.0.

Wickham, H., Hester, J., and Chang, W. (2020). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.3.0.

Wolen, A. and Hartgerink, C. (2020). *osfr: Interface to the 'Open Science Framework' ('OSF')*. R package version 0.2.8.

Conflict of Interest: 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.

frontiers in Neuroinformatics

ORIGINAL RESEARCH
published: 02 December 2021
doi: 10.3389/fninf.2021.738342

Providing Evidence for the Null Hypothesis in Functional Magnetic Resonance Imaging Using Group-Level Bayesian Inference

*Ruslan Masharipov, Irina Knyazeva, Yaroslav Nikolaev, Alexander Korotkov, Michael Didur, Denis Cherednichenko and Maxim Kireev**

N. P. Bechtereva Institute of the Human Brain, Russian Academy of Sciences, Saint Petersburg, Russia

Classical null hypothesis significance testing is limited to the rejection of the point-null hypothesis; it does not allow the interpretation of non-significant results. This leads to a bias against the null hypothesis. Herein, we discuss statistical approaches to 'null effect' assessment focusing on the Bayesian parameter inference (BPI). Although Bayesian methods have been theoretically elaborated and implemented in common neuroimaging software packages, they are not widely used for 'null effect' assessment. BPI considers the posterior probability of finding the effect within or outside the region of practical equivalence to the null value. It can be used to find both 'activated/deactivated' and 'not activated' voxels or to indicate that the obtained data are not sufficient using a single decision rule. It also allows to evaluate the data as the sample size increases and decide to stop the experiment if the obtained data are sufficient to make a confident inference. To demonstrate the advantages of using BPI for fMRI data group analysis, we compare it with classical null hypothesis significance testing on empirical data. We also use simulated data to show how BPI performs under different effect sizes, noise levels, noise distributions and sample sizes. Finally, we consider the problem of defining the region of practical equivalence for BPI and discuss possible applications of BPI in fMRI studies. To facilitate 'null effect' assessment for fMRI practitioners, we provide Statistical Parametric Mapping 12 based toolbox for Bayesian inference.

Keywords: null results, fMRI, Bayesian analyses, human brain, statistical parametric mapping

OPEN ACCESS

Edited by:
Ting Zhao,
Janelia Research Campus,
United States

Reviewed by:
Marc De Kamps,
University of Leeds, United Kingdom
Yikang Liu,
United Imaging Intelligence,
United States

****Correspondence:***
Maxim Kireev
kireev@ihb.spb.ru

Received: *08 July 2021*
Accepted: *05 November 2021*
Published: *02 December 2021*

Citation:
Masharipov R, Knyazeva I, Nikolaev Y, Korotkov A, Didur M, Cherednichenko D and Kireev M (2021) Providing Evidence for the Null Hypothesis in Functional Magnetic Resonance Imaging Using Group-Level Bayesian Inference. Front. Neuroinform. 15:738342. doi: 10.3389/fninf.2021.738342

INTRODUCTION

In the neuroimaging field, it is a common practice to identify statistically significant differences in local brain activity using the general linear model approach for mass-univariate null hypothesis significance testing (NHST) (Friston et al., 1994). NHST considers the probability of obtaining the observed data, or more extreme data, given that the null hypothesis of no difference is true. This probability, or *p*-value, of 0.01, means that, on average, in one out of 100 'hypothetical' replications of the experiment, we find a difference no less than the one found under the null hypothesis. We conventionally suppose that this is unlikely, therefore, we 'reject the null'; that is, NHST employs 'proof by contradiction' (Cohen, 1994). Conversely, when the *p*-value is large, it is tempting to 'accept the null.' However, the absence of evidence is not evidence of absence (Altman and Bland, 1995). Using NHST, we can only state that we have 'failed to reject the null.' Therefore, in the classical NHST framework, the question of interpreting non-significant results remains.

The most pervasive misinterpretation of non-significant results is that they provide evidence for the null hypothesis that there is no difference, or 'no effect' (Nickerson, 2000; Greenland et al., 2016; Wasserstein and Lazar, 2016). In fact, non-significant results can be obtained in two cases (Dienes, 2014): (1) the data are insufficient to distinguish the alternative from the null hypothesis, or (2) an effect is indeed null or trivial. To date, the extent to which the problem of making 'no effect' conclusions from non-significant results have affected the field of neuroimaging remains unclear, particularly in functional magnetic resonance imaging (fMRI) studies[1]. Regarding other fields of science such as psychology, neuropsychology, and biology, it was found that in 38–72% of surveyed articles, the null hypothesis was accepted based on non-significant results only (Finch et al., 2001; Schatz et al., 2005; Fidler et al., 2006; Hoekstra et al., 2006; Aczel et al., 2018).

Not mentioning non-significant results at all is another problem. Firstly, some authors may consider non-significant results disappointing or not worth publishing. Secondly, papers with non-significant results are less likely to be published. This publishing bias is also known as the 'file-drawer problem' (Rosenthal, 1979; Ioannidis et al., 2014; de Winter and Dodou, 2015; for evidence in fMRI studies, see Jennings and Van Horn, 2012; Acar et al., 2018; David et al., 2018; Samartsidis et al., 2020). Prejudice against the null hypothesis systematically biases our knowledge of true effects (Greenwald, 1975).

This problem is further compounded by the fact that NHST is usually based on the point-null hypothesis, that is, the hypothesis that the effect is *exactly* zero. However, the probability thereof is zero (Meehl, 1967; Friston et al., 2002a). This means that studies with a sufficiently large sample size will find statistically significant differences even when the effect is trivial or has no *practical* significance (Cohen, 1965, 1994; Serlin and Lapsley, 1985; Kirk, 1996).

Having the means to assess non-significant results would mitigate these problems. To this end, two main alternatives are available: Firstly, there are frequentist approaches that shift from point-null to interval-null hypothesis testing, for example, equivalence testing based on the two one-sided tests (TOST) procedure (Schuirmann, 1987; Wellek, 2010). Secondly, Bayesian approaches that are based on posterior parameter distributions (Lindley, 1965; Greenwald, 1975; Kruschke, 2010) and Bayes factors (Jeffreys, 1939/1948; Kass and Raftery, 1995; Rouder et al., 2009). The advantage of frequentist approaches is that they do not require a substantial paradigm shift (Lakens, 2017; Campbell and Gustafson, 2018). However, it has been argued that Bayesian approaches may be more natural and straightforward than frequentist approaches (Edwards et al., 1963; Lindley, 1975; Friston et al., 2002a; Wagenmakers, 2007; Rouder et al., 2009; Dienes, 2014; Kruschke and Liddell, 2017b). It has long been noted that we tend to perceive lower *p*-values as stronger evidence for the alternative hypothesis, and higher *p*-values as evidence for the null, i.e., the 'inverse probability' fallacy as it is referred to by Cohen (1994). This is what we obtain in Bayesian approaches by calculating posterior probabilities. Instead of considering infinite 'hypothetical' replications and employing probabilistic 'proof by contradiction,' Bayesian approaches directly provide evidence for the null and alternative hypotheses given the data, updating our prior beliefs in light of new relevant information. Bayesian inference allows us to 'reject and accept' the null hypothesis on an equal footing. Moreover, it allows us to talk about 'low confidence,' indicating the need to either accumulate more data or revise the study design (see **Figure 1**).

Despite the importance of this issue, and the high level of theoretical elaboration and implementation of Bayesian methods in common neuroimaging software programs, for example, Statistical Parametric Mapping 12 (SPM12) and FMRIB's Software Library (FSL), to date, only a few fMRI studies implemented the Bayesian inference to assess 'null effects' (for example, see subject-level analysis in Magerkurth et al., 2015, group-level analysis in Dandolo and Schwabe, 2019; Feng et al., 2019). Therefore, this study is intended to introduce fMRI practitioners to the methods for assessing 'null effects.' In particular, we focus on Bayesian parameter inference (Friston and Penny, 2003; Penny and Ridgway, 2013), as implemented in SPM12. Although Bayesian methods have been described elsewhere, the distinguishing feature of this study is that we aim to demonstrate the practical implementation of Bayesian inference to the assessment of 'null effects,' and reemphasize its contributions over and above those of classical NHST. We deliberately aim to avoid mathematical details, which can be found elsewhere (Genovese, 2000; Friston et al., 2002a, 2007; Friston and Penny, 2003; Penny et al., 2003, 2005, 2007; Penny and Ridgway, 2013; Woolrich et al., 2004). Firstly, we briefly review the frequentist and Bayesian approaches for the assessment of the 'null effects.' Next, we compare the classical NHST and Bayesian parameter inference using the Human Connectome Project (HCP) and the UCLA Consortium for Neuropsychiatric Phenomics datasets, focusing on group-level analysis. We then consider the choice of the threshold of the effect size for Bayesian parameter inference and estimate the typical effect sizes in different fMRI task designs. To demonstrate how the common sources of variability in empirical data influence NHST and Bayesian parameter inference, we examined their behavior for different sample sizes and spatial smoothing. We also used simulated data to assess BPI performance under different effect sizes, noise levels, noise distributions and sample sizes. Finally, we discuss practical research and clinical applications of Bayesian inference.

[1]Here are some examples of 'no effect' conclusions that can be found in the fMRI literature: (a) brain area was not activated, (b) brain area was not involved in the function, (c) no effect was found in the brain area ($p > 0.05$), (d) both groups showed no differences, which can be interpreted as evidence against the alternative hypothesis; (e) patients have similar responses to both conditions ($p > 0.05$), that is, they have difficulties in differentiating these conditions; (f) lack of significant correlation during treatment suggest a protective impact of the therapy on brain areas.

THEORY

In this section, we briefly describe the classical NHST framework and review statistical methods which can be used to assess the 'null effect.' We also considered two historical trends

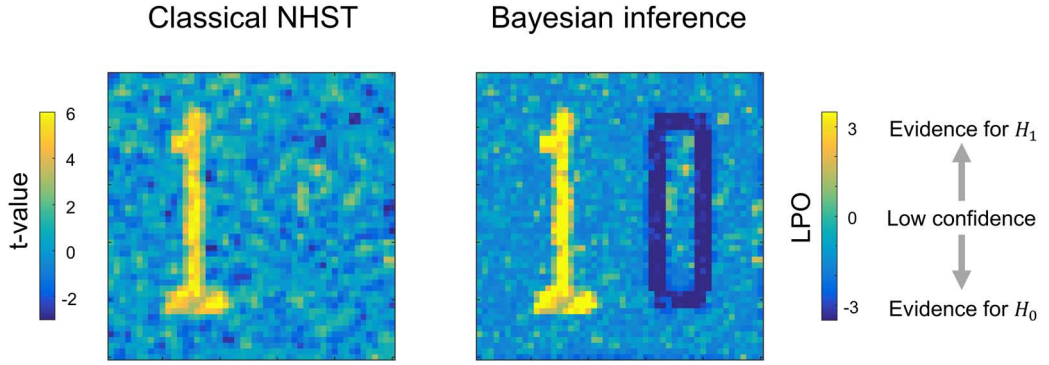


FIGURE 1 | Possible results for the same data, obtained using classical NHST and Bayesian parameter inference. Classical NHST detects only areas with a statistically significant difference ('number one'). Bayesian parameter inference based on the logarithm of posterior probability odds (*LPO*) provides us with additional information that is not available in classical NHST: (1) it provides relative evidence for the null (H_0) and alternative (H_1) hypotheses, (2) it detects areas with a trivial effect size ('number zero'), (3) it indicates 'low confidence' areas surrounding the 'number one' and 'number zero.' To make this conceptual illustration, we generated 100 images consisted of 50 × 50 voxels smoothed by 2 voxel full width at half maximum (FWHM) Gaussian kernel. Data were drawn from normal distributions with different mean, m, and standard deviation, SD. For the 'number one,' m = 0.1, SD = 0.37. For the 'number zero,' m = 0, SD = 0.6. For the 'low confidence' area, m = 0.01, SD = 0.37. *LPOs* were calculated using an effect size threshold of 0.02. The code to recreate the illustration is available online https://github.com/Masharipov/BPI_2021/tree/main/conceptual_illustration.

in statistical analysis: the shift from point-null hypothesis testing to interval estimation and interval-null hypothesis testing (Murphy and Myors, 2004; Wellek, 2010; Cumming, 2013), and the shift from frequentist to Bayesian approaches (Kruschke and Liddell, 2017b).

Classical Null Hypothesis Significance Testing Framework

Most task-based fMRI studies rely on the general linear model approach (Friston et al., 1994; Poline and Brett, 2012). It provides a simple way to separate blood-oxygenated-level dependent (BOLD) signals associated with particular task conditions from nuisance signals and residual noise when analyzing single-subject data (subject-level analysis). At the same time, it allows us to analyze mean BOLD signals within one group of subjects or between different groups (group-level analysis). Firstly, we must specify a general linear model and estimate its parameters:

$$Y = X\beta + \varepsilon \quad (1)$$

where Y are the data (further, D), X is the design matrix, which includes regressors of interest and nuisance regressors, β are the model parameters ('beta values'), and ε is residual noise or error, which is assumed to have a zero-mean normal distribution. At the subject level of analysis, the data are BOLD-signals. At the group level, the data are linear contrasts of parameters estimated at the subject level, which typically reflect individual subject amplitudes of BOLD responses evoked in particular task conditions. In turn, the parameters of the group-level general linear model reflect the group mean BOLD responses evoked in particular task conditions and groups of subjects. The linear contrast of these parameters, $\theta = c\beta$, represents the experimental effect of interest (hereinafter '*the effect*'), expressed as the difference between conditions or groups of subjects.

Next, we test the effect against the point-null hypothesis, H_0: $\theta = \gamma$ (usually, $\theta = 0$). To do this, we use test statistics that summarize the data in a single value, for example, the t-value. For the one-sample case, the t-value is the ratio of the discrepancy of the estimated effect from the hypothetical null value to its standard error. Finally, we calculate the probability of obtaining the observed t-value or a more extreme value, given that the null hypothesis is true (p-value). This is also commonly formulated as the probability of obtaining the observed data or more extreme data, given that the null hypothesis is true (Cohen, 1994). It can be simply written as a conditional probability $P(D+|H_0)$, where '$D+$' denotes the observed data or more extreme data which can be obtained in infinite 'hypothetical' replications under the null (Schneider, 2014, 2018). If this probability is lower than some conventional threshold, or alpha level (for example, $\alpha = 0.05$), then we can 'reject the null hypothesis' and state that we found a statistically significant effect. When this procedure is repeated for a massive number of voxels, it is referred to as 'mass-univariate analysis.' However, if we consider m = 100 000 voxels with no true effect and repeat significance testing for each voxel at $\alpha = 0.05$, we would expect to obtain 5000 false rejections of the null hypothesis (false positives). To control the number of false positives, we must reduce the alpha level for each significance test by applying the multiple comparison correction (Genovese et al., 2002; Nichols and Hayasaka, 2003; Nichols, 2012).

To date, the classical NHST has been the most widely used statistical inference method in neuroscience, psychology, and biomedicine (Szucs and Ioannidis, 2017, 2020; Ioannidis, 2019). It is often criticized for the use of the point-null hypothesis (Meehl, 1967), also known as the 'nil null' (Cohen, 1994) or 'sharp null' hypothesis (Edwards et al., 1963). It was argued that the point-null hypothesis could be appropriate only in hard sciences such as physics, but it is always false in soft sciences; this problem is sometimes known as the Meehl's paradox (Meehl, 1967, 1978; Serlin and Lapsley, 1985, 1993; Cohen, 1994; Kirk, 1996). In the

case of fMRI research, we face complex brain activity which is influenced by numerous psychophysiological factors. This means that with a large amount of data, we find a statistically significant effect in all voxels for any linear contrast (Friston et al., 2002a). For example, Gonzalez-Castillo et al. (2012) showed a statistically significant difference between simple visual stimulation and rest in over 95% of the brain when averaging single-subject data from 100 runs (approximately 8 h of scanning), which consisted of five blocks of stimulation (20 s of visual stimulation, 40 s of rest). Approximately half of the brain areas showed statistically significant positive effects or 'activations,' whereas the other half showed statistically significant negative effects or 'deactivations.'

Whole-brain ''activations/deactivations' can also be found when analyzing large datasets such as the HCP ($N > 1000$) or UK Biobank ($N > 10\,000$) datasets. For example, Smith and Nichols (2018) showed significant positive and negative effects for the emotion processing task ('Emotional faces vs. Shapes' contrast) in 81% of voxels using data from UK Biobank ($N = 12\,600$) and conservative Bonferroni multiple comparison correction. When we increase the sample size, the effect estimate does not change much. Still, the standard error in the denominator of the t-value becomes increasingly smaller, resulting in negligible effects becoming statistically significant. Thus, the classical NHST ignores the magnitude of the effect. Attempts to overcome this problem led to the proposal of making a distinction between 'statistical significance' and 'material significance' (Hodges and Lehmann, 1954) or 'practical significance' (Cohen, 1965; Kirk, 1996). That is, we can test whether the effect size is larger or smaller than some practically meaningful value using interval-null hypothesis testing (Friston et al., 2002a,b; Friston, 2013). In this case, we use the terms 'activations' and 'deactivations' for those voxels that show a practically significant positive or negative effect.

Frequentist Approach to Interval-Null Hypothesis Testing

Interval-null hypothesis testing is widely used in medicine and biology (Meyners, 2012). Consider, for example, a pharmacological study designed to compare a new treatment with an old treatment that has already shown its effectiveness. Let β_{new} be the mean effect on brain activity of the new treatment and β_{old} the mean effect of the old treatment. Then, $\theta = (\beta_{new} - \beta_{old})$ is the relative effect of the new treatment. The practical significance is defined by the effect size (ES) threshold γ. If a larger effect on brain activity is preferable, then we can test whether there is a practically meaningful difference in a positive direction (H_1: $\theta > \gamma$ vs. H_0: $\theta \leq \gamma$). This procedure is known as the *superiority test* (see **Figure 2A**). We can also test whether the effect of the new treatment is no worse (practically smaller) than the effect of the old treatment (H_1: $\theta > -\gamma$ vs. H_0: $\theta \leq -\gamma$). This procedure is sometimes known as the *non-inferiority test* (see **Figure 2B**). If a smaller effect on brain activity is preferable, we can use the superiority or non-inferiority test in the opposite direction (see **Figures 2C,D**). The combination of these two superiority tests allows us to find a practically meaningful difference in both directions (H_1: $\theta > \gamma$ and $\theta < -\gamma$ vs. H_0: $-\gamma \leq \theta \leq \gamma$), that is, the *minimum-effect test* (see **Figure 2E**). The combination of the two non-inferiority tests allows us to reject the hypothesis of practically meaningful differences in any direction (H_1: $-\gamma \leq \theta \leq \gamma$ vs. H_0: $\theta > \gamma$ and $\theta < -\gamma$). This is the most widely used approach to *equivalence testing*, known as the *two one-sided tests* (TOST) procedure (see **Figure 2F**). For more details on the superiority and minimum-effect tests, see Serlin and Lapsley (1985, 1993), Murphy and Myors (1999, 2004). For more details on the non-inferiority test and TOST procedure see Schuirmann (1987), Rogers et al. (1993), Wellek (2010), Meyners (2012), Lakens (2017).

The interval $[-\gamma; \gamma]$ defines trivially small effect sizes that we consider to be equivalent to the 'null effect' for practical purposes. This interval is also known as the 'equivalence interval' (Schuirmann, 1987) or 'region of practical equivalence (ROPE)' (Kruschke, 2011). The TOST procedure, in contrast to classical NHST, allows us to assess the 'null effects.' If we reject the null hypothesis of a practically meaningful difference, we can conclude that the effect is trivially small. The TOST procedure can also be intuitively related to frequentist interval estimates, known as confidence intervals ('confidence interval approach,' Westlake, 1972; Schuirmann, 1987). Confidence intervals reflect the uncertainty in the point estimation of the parameters defined by its standard error. The confidence level of $(1 - \alpha)$ means that among infinite 'hypothetical' replications, $(1 - \alpha)$% of the confidence intervals will contain the true effect under the null. Therefore, the TOST procedure is operationally identical to considering whether the $(1 - 2\alpha)$% confidence interval falls entirely into the ROPE, as it uses two one-sided tests with an alpha level of α.

Interval-null hypothesis testing can be used in fMRI studies not only to compare the effects of different treatments. For example, we can apply superiority tests in the positive and negative directions to detect 'activated' and 'deactivated' voxels and additionally apply the TOST procedure to detect 'not activated' voxels. However, even though we can solve the Meehl's paradox and assess the 'null effects' by switching from point-null to interval-null hypothesis testing within the frequentist approach, this approach still has fundamental philosophical and practical difficulties which can be effectively addressed using Bayesian statistics.

Difficulties of the Frequentist Approach

The pitfalls of the frequentist approach have been actively discussed by statisticians and researchers for decades. Here, we briefly mention a few of the main problems associated with the frequency approach.

(1) NHST is a hybrid of Fisher's approach that focuses on the *p*-value (thought to be a measure of evidence against the null hypothesis), and Neyman-Pearson's approach that focuses on controlling false positives with the alpha level while maximizing true positives in long-run replications. These two approaches are argued to be incompatible and have given rise to several misinterpretations among researchers, for example, confusing the meaning of *p*-values and alpha levels (Edwards et al., 1963; Gigerenzer, 1993; Goodman, 1993; Royall, 1997;

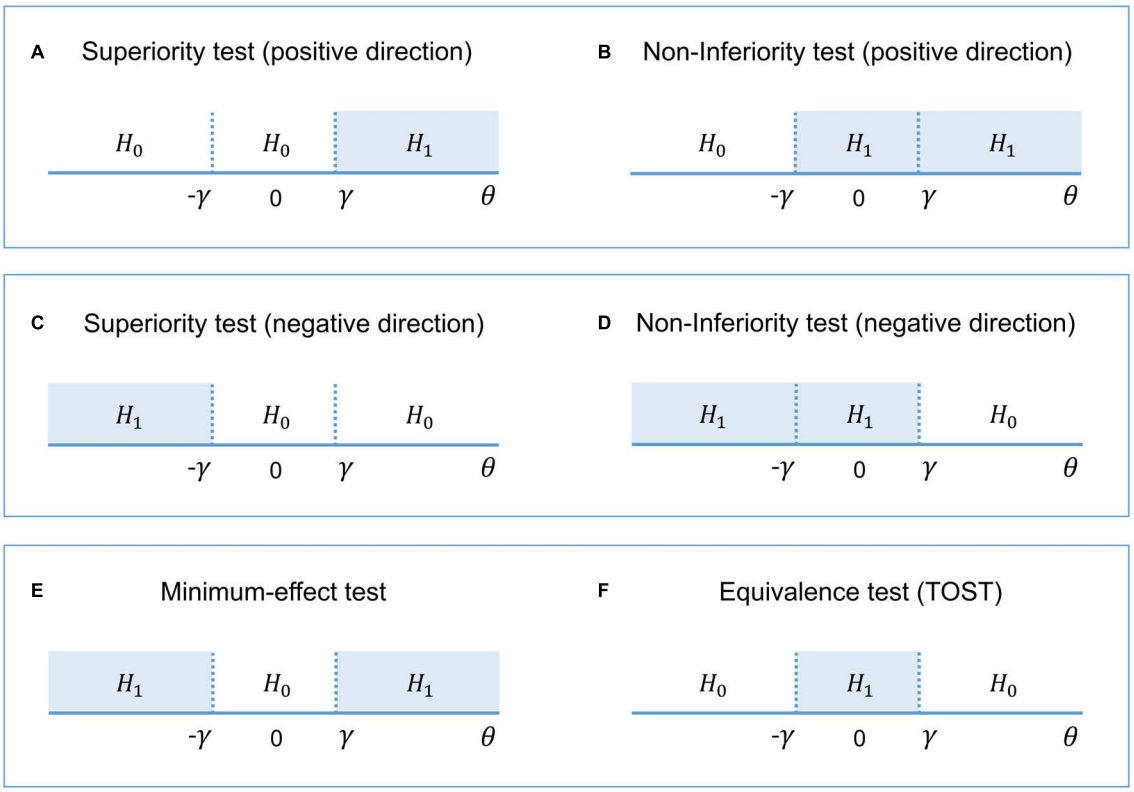


FIGURE 2 | The alternative (H_1) and null (H_0) hypotheses for different types of interval-null hypotheses tests. **(A,B)** One-sided tests in the positive direction ('the larger is better'). **(C,D)** One-sided tests in the negative direction ('the smaller is better'). **(E)** Combination of both superiority tests. **(F)** Combination of both non-inferiority tests.

Finch et al., 2001; Berger, 2003; Hubbard and Bayarri, 2003; Turkheimer et al., 2004; Schneider, 2014; Perezgonzalez, 2015; Szucs and Ioannidis, 2017; Greenland, 2019).

(2) The logical structure of NHST is the same as that of 'proof by contradiction' or 'indirect proof,' which becomes formally invalid when applied to probabilistic statements (Pollard and Richardson, 1987; Cohen, 1994; Falk and Greenbaum, 1995; Nickerson, 2000; Sober, 2008; Schneider, 2014, 2018; Wagenmakers et al., 2017; but see Hagen, 1997). Valid 'proof by contradiction' can be expressed in syllogistic form as: (1) 'If A, then B' (Premise No 1), (2) 'Not B' (Premise No 2), (3) 'Therefore not A' (Conclusion). Probabilistic 'proof by contradiction' in relation to NHST can be formulated as: (1) 'If H_0 is true, then $D+$ are highly unlikely, (2) '$D+$ was obtained,' (3) 'Therefore H_0 is highly unlikely.' This problem is also referred to as the 'illusion of probabilistic proof by contradiction' (Falk and Greenbaum, 1995). To illustrate the fallacy of such logic, consider the following example from Pollard and Richardson (1987): (1) 'If a person is an American (H_0), then he is most probably not a member of Congress,' (2) 'The person is a member of Congress,' (3) 'Therefore the person is most probably not an American.' Based on this, one 'rejects the null' and makes an obviously wrong inference, as only American citizens can be a member of Congress. At the same time, using Bayesian statistics, we can show that the null hypothesis ('the person is an American') is true (see the Bayesian solution of the 'Congress example' in the **Supplementary Materials**). The 'illusion of probabilistic proof by contradiction' leads to widespread confusion between the probability of obtaining the data, or more extreme data, under the null $P(D+|H_0)$ and the probability of the null under the data $P(H_0|\mathrm{D})$ (Pollard and Richardson, 1987; Gigerenzer, 1993; Cohen, 1994; Falk and Greenbaum, 1995; Nickerson, 2000; Finch et al., 2001; Hoekstra et al., 2006; Goodman, 2008; Greenland et al., 2016; Wasserstein and Lazar, 2016; Amrhein et al., 2017). The latter is a posterior probability calculated based on Bayes' rule. The fact that researchers usually treat the *p*-value as a continuous measure of evidence (the Fisherian interpretation) only exacerbates this problem. 'The lower the *p*-value, the stronger the evidence against the null' statement can be erroneously transformed to statements such as 'the lower the *p*-value, the stronger the evidence for the alternative' or 'the higher the *p*-value, the stronger the evidence for the null.' NHST can only provide evidence *against*, but never *for*, a hypothesis. In contrast, posterior probability provides direct evidence for a hypothesis; hence, it has a simple intuitive interpretation.

(3) The *p*-value is not a plausible measure of evidence (Berger and Berry, 1988; Berger and Sellke, 1987; Cornfield, 1966; Goodman, 1993; Hubbard and Lindsay, 2008; Johansson, 2011; Royall, 1986; Wagenmakers, 2007; Wagenmakers et al., 2008, 2017; Wasserstein and Lazar, 2016;

bet see Greenland, 2019). The frequentist approach considers infinite 'hypothetical' replications of the experiment (sampling distribution); that is, the *p*-value depends on unobserved ('more extreme') data. One of the most prominent theorists of Bayesian statistics, Harold Jeffreys, put it as follows: '*What the use of P implies, therefore, is that a hypothesis that may be true may be rejected because it has not predicted observable results that have not occurred*' (Jeffreys, 1939/1948, p. 357). In turn, the sampling distribution depends on the researcher's intentions. These intentions may include different kinds of *multiplicities*, such as multiple comparisons, double-sided comparisons, secondary analyses, subgroup analyses, exploratory analyses, preliminary analyses, and interim analyses of sequentially obtained data with optional stopping (Gopalan and Berry, 1998). Two researchers with different intentions may obtain different *p*-values based on the same dataset. The problem is that these intentions are usually unknown. When null findings are considered disappointing, it is tempting to increase the sample size until one obtains a statistically significant result. However, a statistically significant result may arise when the null is, in fact, true, which can be shown by Bayesian statistics. That is, the *p*-value usually exaggerates evidence against the null hypothesis. The discrepancy that may arise between frequentist and Bayesian inference is also known as the Jeffreys–Lindley paradox (Jeffreys, 1939/1948; Lindley, 1957). In addition, it is argued that a consistent measure of evidence should not depend on the sample size (Cornfield, 1966). However, identical *p*-values provide different evidence against the null hypothesis for small and large sample sizes (Wagenmakers, 2007). In contrast, evidence provided by posterior probabilities and Bayes factors depends only on the exact observed data and the prior, and does not depend on the testing or stopping intentions or the sample size (Wagenmakers, 2007; Kruschke and Liddell, 2017b).

(4) Although frequentist interval estimates (Cohen, 1990, 1994; Cumming, 2013) and interval-based hypothesis testing (Murphy and Myors, 2004; Wellek, 2010; Meyners, 2012; Lakens, 2017) greatly facilitate the mitigation of the abovementioned pitfalls in data interpretation, they are still subject to some of the same types of problems as the *p*-values and classic NHST (Cortina and Dunlap, 1997; Nickerson, 2000; Belia et al., 2005; Wagenmakers et al., 2008; Hoekstra et al., 2014; Morey et al., 2015; Greenland et al., 2016; Kruschke and Liddell, 2017a). Confidence intervals also depend on unobserved data and the intentions of the researcher. Moreover, the meaning of confidence intervals seems counterintuitive to many researchers. For example, one of the most common misinterpretations of the (1 – α)% confidence interval is that the probability of finding an effect within the confidence interval is (1 – α)%. In fact, it is a Bayesian interval estimate known as a *credible* interval.

Nevertheless, we would like to emphasize that we do not advocate abandoning the frequency approach. Correctly interpreted frequentist interval-based hypothesis testing with *a priori* power analysis defining the sample size and proper multiplicity adjustments often lead to conclusions similar to those of Bayesian inference (Lakens et al., 2018). However, it may be logically and practically difficult to carry out an appropriate power analysis and make multiplicity adjustments (Berry and Hochberg, 1999; Cramer et al., 2015; Streiner, 2015; Schönbrodt et al., 2017; Sjölander and Vansteelandt, 2019). These procedures may be even more complicated in fMRI research than in psychological or social studies (see discussion on power analysis in Mumford and Nichols, 2008; Joyce and Hayasaka, 2012; Mumford, 2012; Cremers et al., 2017; Poldrack et al., 2017; multiple comparisons in Nichols and Hayasaka, 2003; Nichols, 2012; Eklund et al., 2016; and other types of multiplicities in Turkheimer et al., 2004; Chen et al., 2018, 2019, 2020; Alberton et al., 2020). For example, at the beginning of a long-term study, one may want to check whether stimulus onset timings are precisely synchronized with fMRI data collection and perform preliminary analysis on the first five subjects. The question of whether the researcher must make an adjustment for this technical check when reporting the results for the final sample become important in the frequentist approach. Such preliminary analyses (or other forms of interim analyses) are generally not considered a source of concern in Bayesian inference because posterior probabilities do not depend on the sampling plan (for discussion, see Berry, 1988; Berger and Berry, 1988; Edwards et al., 1963; Wagenmakers, 2007; Kruschke and Liddell, 2017b; Rouder, 2014; Schönbrodt et al., 2017). Or, for example, one may want to find both 'activated/deactivated' and 'not activated' brain areas and use two superiority tests in combination with the TOST procedure. It is not trivial to make appropriate multiplicity adjustments in this case. In contrast, Bayesian inference suggests a single decision rule without the need for additional adjustments. Moreover, to our knowledge, practical implementations of superiority tests and the TOST procedure in common software for fMRI data analysis do not yet exist. At the same time, Bayesian analysis has already been implemented in SPM12[2] and is easily accessible to end-users. It consists of two steps: Bayesian parameter estimation and Bayesian inference. In general, it is not necessary to use Bayesian analysis at the subject level of analysis to apply it at the group level. One can combine computationally less demanding frequentist parameter estimation for single subjects with Bayesian estimation and inference at the group level. In the next sections, we consider the group-level Bayesian analysis implemented in SPM12.

Bayesian Parameter Estimation

Bayesian statistics is based on Bayes' rule:

$$P(H|D) = \frac{P(D|H)\,P(H)}{P(D)} \tag{2}$$

where $P(H|D)$ is the probability of the hypothesis given the obtained data or posterior probability. $P(D|H)$ is the probability of obtaining the *exact* data given the hypothesis or the likelihood (notice the difference from $P(D+|H)$, which includes *more extreme* data). $P(H)$ is the prior probability of the hypothesis (our knowledge of the hypothesis before we obtain the data). $P(D)$ is a normalizing constant ensuring that the sum of posterior probabilities over all possible hypotheses equals one (marginal likelihood). In the case of mutually exclusive hypotheses, the denominator of Bayes's rule is the

[2]https://www.fil.ion.ucl.ac.uk/spm/software/spm12

sum of the probabilities of obtaining the data under any of the possible hypotheses, multiplied by its prior probability. For example, if we consider two mutually exclusive hypotheses H_0 and H_1, then $P(D) = P(D|H_0)\ P(H_0) + P(D|H_1)P(H_1)$ and $P(H_0|D) + P(H_1|D) = 1$. When we consider continuous hypotheses, the denominator is obtained by integrating over all hypotheses (parameter spaces). For relatively simple models, these integrals can be solved analytically. However, for more complex models, the integrals become analytically intractable. In this case, there are two main approaches to obtain the posterior probability: (1) use computationally demanding numerical integration (Markov chain Monte Carlo methods); (2) use less accurate but computationally efficient analytical approximations to the posterior distribution (e.g., Expectation Maximization or Variational Bayes techniques). Describing these procedures go beyond the scope of this paper and described elsewhere (for their implementations in fMRI analysis, see Genovese, 2000; Friston et al., 2002a, 2007; Friston and Penny, 2003; Penny et al., 2003, 2005, 2007; Penny and Ridgway, 2013; Woolrich et al., 2004).

In verbal form, Bayes' rule can be expressed as:

$$Posterior \propto Likelihood \times Prior$$

This means that we can update our prior beliefs about the hypothesis based on the obtained data.

One of the main difficulties in using Bayesian statistics, in addition to the computational complexity, is the choice of appropriate prior assumptions. The prior can be chosen based on theoretical arguments or from independent experimental data (full Bayes approach). At the same time, if the data are organized hierarchically, which is the case for neuroimaging data, priors can be specified based on the obtained data itself using an empirical Bayes approach. The lower level of the hierarchy corresponds to the experimental effects at any given voxel, and the higher level of the hierarchy comprises the effect over all voxels. Thus, the variance of the experimental effect over all voxels can be used as the prior variance of the effect at any given voxel. This approach is known as the parametric empirical Bayes (PEB) with the 'global shrinkage' prior (Friston and Penny, 2003). The prior variance is estimated from the data under the assumption that the prior probability density corresponds to a Gaussian distribution with zero mean. In other words, a global experimental effect is assumed to be absent. An increase in local activity can be detected in some brain areas; a decrease can be found in others, but the total change in neural metabolism in the whole brain is approximately zero. This is a reasonable physiological assumption because studies of brain energy metabolism have shown that the global metabolism is 'remarkably constant despite widely varying mental and motoric activity' (Raichle and Gusnard, 2002), and 'the changes in the global measurements of blood flow and metabolism' are 'too small to be measured' by functional imaging techniques such as PET and fMRI (Gusnard and Raichle, 2001).

Now, we can rewrite Bayes' rule (eq. 2) for the effect $\theta = c\beta$:

$$P(\theta \mid D) = \frac{P(D \mid \theta)\, P(\theta)}{P(D)} \tag{3}$$

In the process of Bayesian updating with the 'global shrinkage' prior, the effect estimate 'shrinks' toward zero. The greater the uncertainty of the effect estimate (variability) in a particular voxel, the less confidence in this estimate, and the more it shrinks (see **Figure 3**).

The assumption of a Gaussian prior, likelihood, and posterior essentially reduces computational demands for Bayesian analysis. However, the normality assumption can be violated for empirical data. For example, violations can be observed in the presence of outliers, particularly with small sample sizes or unbalanced designs, which diminishes the validity of the statistical analysis. This problem is not specific to Bayesian analysis but is inherent to all group-level analyses that assume a normal distribution of the effect. Nevertheless, in fMRI studies, the most common approach is to use the Gaussian general linear models (Poline and Brett, 2012), which have been shown to be robust against violations of the normality assumption (Knief and Forstmeier, 2021). Still, we need to be ensured that these assumptions are not violated substantially. If that is the case, one can use Bayesian estimation based on non-Gaussian distributions. In this work, we consider Bayesian estimation with Gaussian 'global shrinkage' prior implemented in SPM12.

After Bayesian parameter estimation, we can apply one of the two main types of Bayesian inference (Penny and Ridgway, 2013): *Bayesian parameter inference (BPI)* or *Bayesian model inference (BMI)*. BPI is also known as Bayesian parameter estimation (Kruschke and Liddell, 2017b). However, we deliberately separate these two terms, as they correspond to two different steps of data analysis in SPM12. BMI is also known as Bayesian model comparison, Bayesian model selection, or Bayesian hypothesis testing (Kruschke and Liddell, 2017b). We chose the term BMI as it is consonant with the term BPI.

Bayesian Parameter Inference

The BPI is based on the posterior probability of finding the effect within or outside the ROPE. Let effects larger than the ES threshold γ be 'activations,' those smaller than $-\gamma$ be 'deactivations,' and those falling within the ROPE $[-\gamma; \gamma]$ be 'no activations.' Then, we can classify voxels as 'activated,' 'deactivated,' or 'not activated' if:

$$P_{act} = P(\theta > \gamma \mid D) \geq P_{thr} \tag{4.1}$$

$$P_{deact} = P(\theta < -\gamma \mid D) \geq P_{thr} \tag{4.2}$$

$$P_{null} = P(-\gamma \leq \theta \leq \gamma \mid D) \geq P_{thr} \tag{4.3}$$

where P_{thr} is the posterior probability threshold (usually $P_{thr} = 95\%$). Note that $P_{act} + P_{deact} + P_{null} = 1$.

If none of the above criteria are satisfied, the data in a particular voxel are insufficient to distinguish voxels that are 'activated/deactivated' from those that are 'not activated.' Hereinafter, we refer to them as 'low confidence' voxels (Magerkurth et al., 2015). This decision rule is also known as the 'ROPE-only' rule (Kruschke and Liddell, 2017a), see also Greenwald (1975); Wellek (2010), Liao et al. (2019). To the

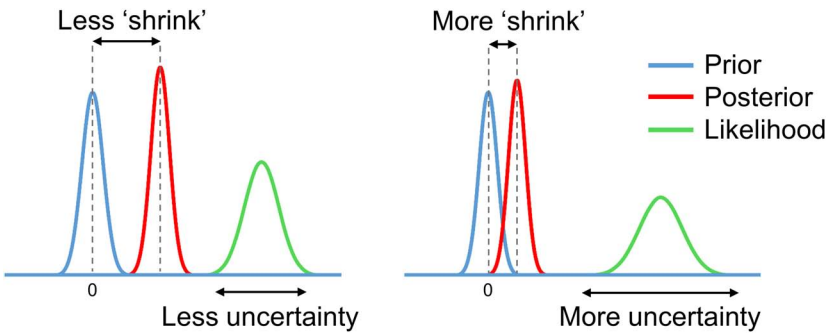


FIGURE 3 | Schematic of Bayesian updating with the 'global shrinkage' prior.

best of our knowledge, the application of this decision rule to neuroimaging data was pioneered by Friston et al. (2002a; 2002b; Friston and Penny, 2003). For convenience and visualization purposes, we can use the natural logarithm of the posterior probability odds (LPO), for example:

$$LPO_{null} = ln\left(\frac{P_{null}}{P_{act} + P_{deact}}\right) = ln\left(\frac{P_{null}}{1 - P_{null}}\right) \quad (5)$$

This allows us to more effectively discriminate voxels with a posterior probability close to unity (Penny and Ridgway, 2013). $LPO_{null} > 3$ corresponds to $P_{null} > 95\%$. In addition, *LPO* also allows us to identify the connection between BPI and BMI. The maps of the *LPO* are termed posterior probability maps (PPMs) in SPM12.

Another possible decision rule considers the overlap between ROPE and the 95% highest density interval (HDI). HDI is a type of credible interval (Bayesian analog of the confidence interval), which contains only the effects with the highest posterior probability density. If the HDI falls entirely inside the ROPE, we can classify voxels as 'not activated.' In contrast, if the HDI lies completely outside the ROPE, we can classify voxels as either 'activated' or 'deactivated.' If the HDI overlaps with the ROPE, we cannot make a confident decision (we can consider them to be 'low confidence' voxels). This decision rule is known as the 'HDI+ROPE' rule (Kruschke and Liddell, 2017a). It is more conservative than the 'ROPE-only' rule because it does not consider the effects from the low-density tails of the posterior probability distribution. Differences between the 'HDI+ROPE' rule and the 'ROPE-only' are most evident for strongly skewed distributions. In such cases, the ROPE may contain more than 95% of the posterior probability distribution, but the 95% HDI may overlap with the ROPE. In the case of a Gaussian posterior probability distribution, both decision rules should produce similar results. The 'HDI+ROPE rule is advocated by Kruschke and Liddell (2017a) and the 'ROPE-only' rule is preferred by Friston et al. (2002a; 2002b; Friston and Penny, 2003), Wellek (2010); Liao et al. (2019). These decision rules are illustrated in **Figure 4**.

Bayesian Model Inference

With BPI, we consider the posterior probabilities of the linear contrast of parameters $\theta = c\beta$. Instead, we can consider models using BMI.

Let H_{alt} and H_{null} be two non-overlapping hypotheses represented by models M_{alt} and M_{null}. These models are defined by two parameter spaces: (1) M_{alt}: $\theta > \gamma$ and $\theta < -\gamma$, and (2) M_{null}: $-\gamma \leq \theta \leq \gamma$.

Now, we can rewrite Bayes' rule (eq. 2) for M_{alt} and M_{null}

$$P(M_{alt} \mid D) = \frac{P(D \mid M_{alt})\, P(M_{alt})}{P(D)} \quad (6.1)$$

$$P(M_{null} \mid D) = \frac{P(D \mid M_{null})\, P(M_{null})}{P(D)} \quad (6.2)$$

If we divide equation (6.1) by (6.2), $P(D)$ is canceled out, and we obtain:

$$\frac{P(M_{alt} \mid D)}{P(M_{null} \mid D)} = \frac{P(D \mid M_{alt})}{P(D \mid M_{null})} \frac{P(M_{alt})}{P(M_{null})} \quad (7)$$

In verbal form equation (7) can be expressed as:

Posterior Odds = Bayes Factor × Prior Odds

The Bayes factor (*BF*) is a multiplier that converts prior model probability odds to posterior model probability odds. It indicates the relative evidence for one model against another. For example, if $BF_{null} = \frac{p(D|M_{null})}{p(D|M_{alt})} = 2$, then the observed data are twice as likely under the null model than under the alternative.

A connection exists between the BPI (eq. 3–5), and BMI (eq. 7) (see Morey and Rouder, 2011; Liao et al., 2019):

$$BF_{null} = \left(\frac{P(-\gamma \leq \theta \leq \gamma|D)}{1 - P(-\gamma \leq \theta \leq \gamma|D)}\right)\left(\frac{1 - P(-\gamma \leq \theta \leq \gamma)}{P(-\gamma \leq \theta \leq \gamma)}\right) \quad (8)$$

or, in verbal form:

$$BF(ROPE)_{null} = \frac{Posterior(\theta \in ROPE)}{Posterior\ (\theta \notin ROPE)} \frac{Prior(\theta \notin ROPE)}{Prior(\theta \in ROPE)}$$

For convenience, *BF* may also be expressed in the form of a natural logarithm:

$$LogBF(ROPE)_{null} = LPO_{null} + ln\left(\frac{Prior(\theta \notin ROPE)}{Prior(\theta \in ROPE)}\right) \quad (9)$$

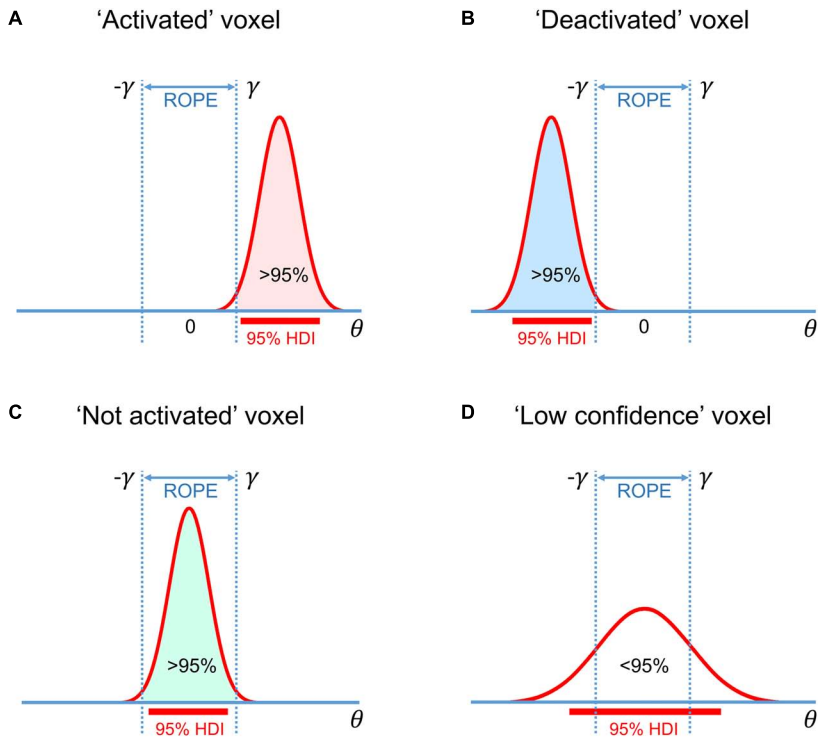


FIGURE 4 | Possible variants of the posterior probability distributions of the effect θ = cβ in **(A)** 'activated' voxels, **(B)** 'deactivated' voxels, **(C)** 'not activated' voxels, **(D)** 'low confidence' voxels. The 'ROPE only' rule considers only the colored parts of the distributions. The 'HDI+ROPE' rule considers overlap between the ROPE and 95% HDI.

$$logBF(ROPE)_{null} \propto LPO_{null} \quad (10)$$

The calculation of *BF* may be computationally challenging, as it requires integration over the parameter space. However, if the ROPE has zero width (point-null hypothesis), then the *BF* has an analytical solution known as the Savage–Dickey ratio (SDR) (Wagenmakers et al., 2010; Friston and Penny, 2011; Rosa et al., 2012; Penny and Ridgway, 2013). $BF(SDR)_{null}$ is calculated by dividing the prior probability density by the posterior probability density at θ = 0. The interpretation of the $BF(SDR)_{null}$ is simple: if the effect size is less likely to equal zero after obtaining the data than before, then $BF(SDR)_{null} < 1$: that is, we have more evidence for M_{alt}. See schematic illustration of BMI based on interval-null and point-null hypotheses and its relation to BPI in **Figure 5**.

Relations Between Frequentist and Bayesian Approaches

Now we can point out the conceptual links between the frequentist and Bayesian approaches.

(1) **Parameter estimation**: When we have no prior information, that is, all parameter values are *a priori* equally probable ('flat' prior), the PEB estimation reduces to the frequentist parameter estimation (maximum likelihood estimation; Friston et al., 2002a).

(2) **Multiplicity adjustments**: One of the major concerns in frequentist inference is the multiplicity problem. In general, after the Bayesian parameter estimation, it is not necessary to classify any voxel as 'activated/deactivated ' or 'not activated.' If we consider *unthresholded* maps of posterior probabilities, *LPOs*, or *LogBFs*, the multiple comparisons problem does not arise (Friston and Penny, 2003). However, if we apply a decision rule to classify voxels, we should control for wrong decisions across multiple comparisons (Woolrich et al., 2009, see also possible loss functions in Muller et al., 2006; Kruschke and Liddell, 2017a). The advantage of PEB with the 'global shrinkage' prior is that it automatically accounts for multiple comparisons without the need for *ad hoc* multiplicity adjustments (Berry, 1988; Friston and Penny, 2003; Gelman et al., 2012). The frequentist approach processes every voxel independently, whereas the PEB algorithm considers joint information from all voxels. Frequentist inference uncorrected for multiple independent comparisons is prone to label noise-driven, random extremes as 'statistically significant.' Bayesian analysis specifies that extreme values are unlikely *a priori*, and thus they shrink toward a common mean (Lindley, 1990; Westfall et al., 1997; Berry and Hochberg, 1999; Friston et al., 2002a,b; Gelman et al., 2012; Kruschke and Liddell, 2017b). If we consider *thresholded* maps of posterior probabilities, for example, $P_{act} > 95\%$, then as many as 5% of 'activated' voxels could be falsely labeled so. This is conceptually similar to the false discovery rate (FDR) correction (Berry and Hochberg, 1999; Friston et al., 2002b; Friston and Penny, 2003; Storey, 2003; Muller et al., 2006; Schwartzman et al., 2009). In practice, BPI with γ = 0 should produce similar results (in terms of the number

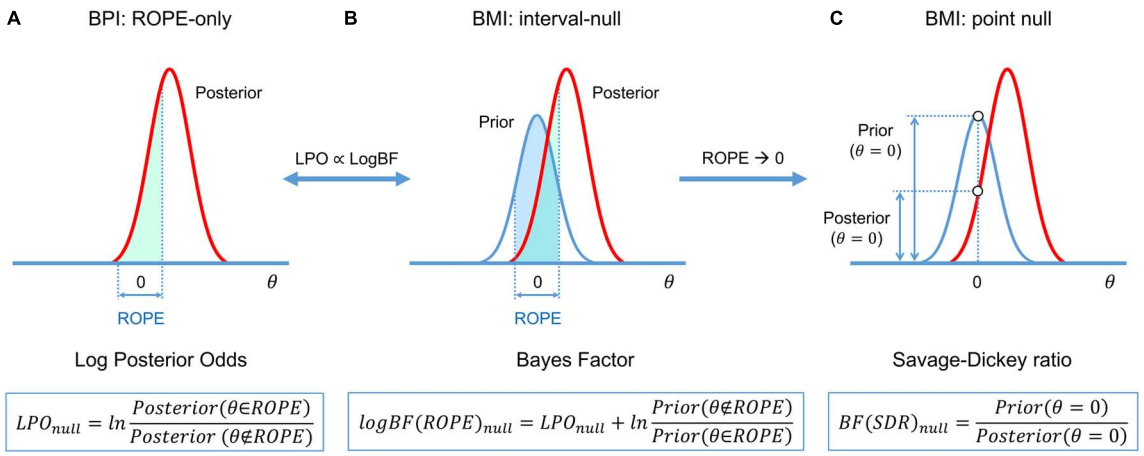


FIGURE 5 | Schematic of *BFs* used in BMI and their relation to *LPO* used in BPI. **(A)** BPI based on the 'ROPE-only' decision rule. **(B)** *BF(ROPE)* is related to the areas under the functions of the posterior and prior probability densities inside and outside the ROPE. **(C)** *BF(SDR)* is the relation between the posterior and prior probability at θ = 0. *LPOs* and *BFs* provide relative evidence for the null and alternative hypotheses.

of 'activated/deactivated' voxels) as classical NHST with FDR correction. If we increase the ES threshold, fewer voxels will be classified as 'activated/deactivated,' and at some γ value, BPI will produce results similar to the more conservative Family Wise Error (FWE) correction[3].

(3) **Interval-based hypothesis testing**: Frequentist interval-based hypothesis testing is conceptually connected with BPI, particularly, the 'HDI+ROPE' decision rule. The former considers the intersection between ROPE and the confidence intervals. The latter considers the intersection between ROPE and the HDI (credible intervals).

(4) **BPI and BMI**: BMI based on *BF(ROPE)* is conceptually linked to BPI based on the 'ROPE-only' decision rule. The interval-based Bayes factor *BF(ROPE)* is proportional to the posterior probability odds. When ROPE is infinitesimally narrow, *BF* can be approximated using the *SDR*. Note that even though *BF(SDR)* is based on the point-null hypothesis, it can still provide evidence for the null hypothesis, in contrast to BPI with γ = 0. However, *BF(SDR)* in PEB settings has not yet been tested using empirical fMRI data. Because the point-null hypothesis is always false (Meehl, 1967), BPI and *BF(ROPE)* may be preferred over *BF(SDR)*.

Definition of the Effect Size Threshold

The main difficulty in applying interval-based methods is the choice of the ES threshold γ. To date, only a few studies have been devoted to determining the minimal relevant effect size. One of them suggested a method to objectively define γ at the subject level of analysis which was calibrated by clinical experts and may be implemented for pre-surgical planning (Magerkurth et al., 2015). At the same time, the problem of choosing the ES threshold γ for the group-level Bayesian analysis remains unresolved.

[3]FDR correction controls the rate of false discoveries (false positives in frequentist terminology) among all significant voxels. FWE correction controls the rate of any false positives in the whole brain.

Several ways in which to define the ES threshold are available. Firstly, we can conduct a pilot study to determine the expected effect sizes. Secondly, we can use data from the literature to determine the typical effect sizes for the condition of interest. Thirdly, we can use the default ES thresholds that are commonly accepted in the field. One of the first ES thresholds proposed in the neuroimaging literature was γ = 0.1% (Friston et al., 2002b). This is the default ES threshold for the subject-level BPI in SPM12. For the group-level BPI, the default ES threshold is one prior standard deviation of the effect γ = 1 *prior* SD_θ (Friston and Penny, 2003). Fourthly, γ can be selected in such a way as to ensure maximum similarity of the activation patterns revealed by classical NHST and Bayesian inference. This would allow us to reanalyze the data using Bayesian inference, reveal similar activation patterns as was previously the case for classic inference, and detect the 'not activated' and 'low confidence' voxels. Lastly, we can consider the posterior probabilities at multiple ES thresholds or compute the ROPE maps (see below).

The ES threshold can be expressed as unstandardized (raw β values or percent signal change) and standardized values (for example, Cohen's d). Raw β values calculated by SPM12 at the first level of analysis represent the BOLD signal in arbitrary units. However, they can be scaled to a more meaningful unit, the BOLD percent signal change (PSC) (Poldrack et al., 2011; Chen et al., 2017). Unstandardized and standardized values have disadvantages and advantages. Different ways exist in which to scale β values to PSC (Pernet, 2014; Chen et al., 2017), which is problematic when comparing the results of different studies. Standardized values represent the effect size in terms of the standard deviation units, which supposedly facilitate the comparison of results between different experiments. However, standardized values are relatively more unstable between measurements and less interpretable (Baguley, 2009; Chen et al., 2017). Moreover, Cohen's d is closely related to the t-value (for one sample case, $d = t/\sqrt{N}$) and may share some drawbacks with t-values. Reimold et al. (2005) showed that spatial smoothing

has a nonlinear effect on voxel variance. Using t-values or Cohen's d for inference in neuroimaging may lead to spatially inaccurate results (spatial shift of local maxima in t-maps or Cohen's d maps compared to PSC-maps). In this study, we focused on PSCs.

It is also important to note that effect sizes (both BOLD PSC and Cohen's d) depend on the MRI scanner (e.g., field strength, coil sensitivity), acquisition parameters (e.g., echo time, spin echo vs. gradient echo sequences) and field inhomogeneity (UIudag et al., 2009). For example, the effect sizes may be underestimated in brain areas near air–tissue interfaces because of field inhomogeneities. This fact further complicates the selection of the ES threshold. However, this does not mean that we should ignore the effect size and return to the point-null hypothesis. One may choose different ES thresholds for different regions of interest, scanners or acquisition parameters.

METHODS

Datasets

Seven block-design tasks were considered from the HCP dataset, including working memory, gambling, motor, language, social cognition, relation processing, and emotion processing tasks (Barch et al., 2013). Two event-related tasks, including the stop-signal and task-switching tasks were considered from the UCLA dataset (Poldrack et al., 2016). The length, conditions, and number of scans of the tasks are provided in the **Supplementary Materials** (**Supplementary Table 1**). A subset of 100 unrelated subjects (S1200 release) was selected from the HCP dataset (54 females, 46 males, mean age = 29.1 ± 3.7 years) for assessment. A total of 115 subjects from the UCLA dataset were included in the analysis (55 females, 60 males, mean age = 31.7 ± 8.9 years) after removing subjects with no data for the stop-signal task, a high level (>15%) of errors in the Go-trials, and those of which the raw data were reported to be problematic (Gorgolewski et al., 2017). See the fMRI acquisition parameters in the **Supplementary Materials**, Par. 1.

Preprocessing

The minimal preprocessing pipelines for the HCP and UCLA datasets were described by Glasser et al. (2013) and Gorgolewski et al. (2017), respectively. Spatial smoothing was applied to the preprocessed images with a 4 mm full width at half maximum (FWHM) Gaussian smoothing kernel. Additionally, to compare the extent to which the performance of classical NHST and BPI depended on the smoothing, we applied 8 mm FWHM smoothing to the emotion processing task. Spatial smoothing was performed using SPM12. The results are reported for the 4 mm FWHM smoothing filter, unless otherwise specified.

Parameter Estimation

Frequentist parameter estimation was applied at the subject level of analysis. A detailed description of the general linear models for each task design is available in the **Supplementary Materials**, Par. 2. Fixation blocks and null events were not modeled explicitly in any of the tasks. Twenty-four head motion regressors were included in each subject-level model (six head motion parameters, six head motion parameters one time point before, and 12 corresponding squared items) to minimize head motion artifacts (Friston et al., 1996). Raw β values were converted to PSC relative to the mean whole-brain 'baseline' signal (**Supplementary Materials**, Par. 3). The linear contrasts of the β values were calculated to describe the effects of interest $\theta = c\beta$ in different tasks. The sum of positive terms in the contrast vector, *c*, is equal to one. The list of contrasts calculated in the current study to explore typical effect sizes is presented in **Supplementary Table 1**. At the group level of analysis, the Bayesian parameter estimation with the 'global shrinkage' prior was applied using SPM12 (v6906). We performed a one-sample test on the linear contrasts created at the subject level of analysis.

Classical Null Hypothesis Significance Testing and Bayesian Parameter Inference

Classical inference was performed using voxel-wise FWE correction with $\alpha = 0.05$. This is the default SPM threshold and is known to be conservative and to guarantee protection from false positives (Eklund et al., 2016). Although voxel-wise FWE correction may be too conservative for small sample sizes, it is recommended when large sample sizes are available (Woo et al., 2014).

Bayesian parameter inference, accessible via the SPM12 GUI, allows the user to declare only whether the voxels are 'activated' or 'deactivated.' The classification of voxels as being either 'not activated' or 'low confidence' requires the posterior mean and variance. At the group level of analysis, SPM12 does not save the posterior variance image. However, the posterior variance can be reconstructed from the image of the noise hyperparameter using a first-order Taylor series approximation (Penny and Ridgway, 2013). Therefore, in the current study, BPI was performed using the developed SPM12-based toolbox[4]. For the 'ROPE-only' rule, the posterior probability threshold was $P_{thr} = 95\%$ ($LPO > 3$). For the 'HDI+ROPE' rule, we used the 95% HDI.

We compared the number of 'activated' voxels (as a percentage of the total number of voxels) detected by Bayesian and classical inference. We also compared the number of 'activated,' 'deactivated,' and 'not activated' voxels detected using BPI with the 'ROPE-only' and 'HDI+ROPE' decision rules and different ES thresholds. To estimate the influence of the sample size on the results, all the above-mentioned analyses were performed with samples of different sizes: 5 to 100 subjects from the HCP dataset (the emotion processing task, 'Emotion > Shape' contrast) and 5 to 115 subjects from the UCLA dataset (the stop signal task, 'Correct Stop > Go' contrast), in steps of 5 subjects. Ten random groups were sampled for each step.

Effect Size Thresholds

We considered three ES thresholds: firstly, the default ES threshold for the subject-level $\gamma = 0.1\%$ (BOLD PSC); secondly, the default ES threshold for the group-level $\gamma = 1$ *prior* SD_θ; thirdly, the $\gamma(Dice_{max})$ threshold, which ensures maximum

[4]https://github.com/Masharipov/Bayesian_inference

similarity of the activation patterns revealed by classical NHST and BPI. The similarity was assessed using the Dice coefficient:

$$Dice\,(\gamma) = \frac{2 * V_{overlap}\,(\gamma)}{V_{classic} + V_{bayesian}\,(\gamma)} \quad (11)$$

where $V_{classic}$ is the number of 'activated' voxels detected using classical NHST, $V_{bayesian}\,(\gamma)$ is the number of 'activated' voxels detected using BMI with the ES threshold γ, and $V_{overlap}$ is the number of 'activated' voxels detected by both methods. A Dice coefficient of 0 indicates no overlap between the patterns, and 1 indicates complete overlap. Dice coefficients were calculated for γ ranging from 0 to 0.4% in steps of 0.001%.

Estimation of Typical Effect Sizes

In the current study, we aimed to provide a reference set of typical effect sizes for different task designs (block and event-related) and different contrasts ('task-condition > control-condition,' 'task-condition > baseline,') in a set of *a priori* defined regions of interest (ROI). Effect sizes were expressed in PSC and Cohen's d. ROI masks were defined using anatomical and *a priori* functional masks. For more details, see **Supplementary Materials**, Par. 4.

Evaluating Bayesian Parameter Inference on Contrasts With No Expected Practically Significant Difference

Bayesian parameter inference should be able to detect the 'null effect' in the majority of voxels when comparing samples with no expected practically significant difference. For example, there may be two groups of healthy adult subjects performing the same task or two sessions with the same task instructions. To test this, we used fMRI data from the emotion processing task. To emulate two 'similar' *independent* samples, 100 healthy adult subjects' contrasts ('Emotion > Shape') were randomly divided into two groups of 50 subjects. A two-sample test comparing the 'Group #1' and 'Group #2' was performed with the assumption of unequal variances between the groups (SPM12 default option). To emulate 'similar' *dependent* samples, we randomized 'Emotion > Shape' contrasts from right-to-left (RL) and left-to-right (LR) phase encoding sessions in the 'Session #1' and 'Session #2' samples. Each sample consisted of 50 contrasts from the RL session and 50 from the LR session. A paired test designed to compare 'Session #1' and 'Session #2' was equivalent to the one-sample test on 50 'RL > LR session' and 50 'LR > RL session' contrasts.

Normality Check

To check for violations of the normality assumption we performed Shapiro-Wilk test (Shapiro and Wilk, 1965) for each voxel for one block-design task ('Emotion >Shape' contrast) and one event-related task ('Correct Stop > Go' contrast). We reported the number of voxels that were significantly non-Gaussian (using $\alpha = 0.001$ uncorrected for multiple comparisons and $\alpha = 0.05$ with Bonferroni correction). We also calculated median kurtosis and skewness across voxels. Kurtosis is a measure of the heaviness of the tails. Skewness is a measure of asymmetry of distribution.

Simulations

The main limitation of using empirical data to assess the performance of statistical methods lies in the lack of knowledge of the ground truth. Therefore, we performed group-level simulations to better understand how the sample size and effect size threshold affect BPI results given different known effect sizes and noises. Simulations also allowed us to assess the robustness of BPI to the violations of the normality assumption. We generated the parameter maps (contrast images) similar to Nichols and Hayasaka (2003); Schwartzman et al. (2009) and Cremers et al. (2017). Each contrast image consisted of 'activated' and 'deactivated' voxels and 'trivial' background voxels surrounding them. Locations of 'activated' and 'deactivated' voxels were specified based on the NeuroSynth meta-analysis results (Yarkoni et al., 2011) obtained using the search terms 'task' and 'default,' respectfully (association test, $\alpha = 0.01$ with FDR correction). Data were drawn from the Pearson system distribution (Johnson et al., 1994) with kurtosis, $Ku = 2.2, 3, 7$ and skewness, $Sk = -0.7, 0, 0.7$. The normal distribution corresponds to $Ku = 3$ and $Sk = 0$. Other combinations of Ku and Sk resulted in four-parameter beta distributions. The mean effect in practically significant ('activated' and 'deactivated') voxels was $\theta = \pm 0.1, 0.2, 0.3\%$. For practically non-significant or 'trivial' voxels, the mean effect was $\theta = \pm 0.04\%$, which can be considered equivalent to the null value for practical purposes ('not activated' voxels). Noise standard deviation was $SD = 0.2, 0.3, 0.4\%$. The mean effect size and noise were consistent with those observed in the empirical data (see **Supplementary Tables 11–19**). Contrast-to-noise ratio was varied from 0.25 to 1.5. For each combination of the Pearson system distribution parameters, we generated 1000 images.

To evaluate sample size dependencies, we randomly drawn images from the full sample ($N = 1000$) ranging from $N = 10$ to 100 (with step 10) and from $N = 150$ to 500 (step 50). This procedure was repeated ten times for each step. The analysis was limited to the single axial slice ($z = 36$ mm) containing 579 'activated' voxels, 500 'deactivated' voxels and 3067 'trivial' or 'not activated' voxels. For classical NHST and BPI, we calculated the number of 'activated' voxels in relation to the total number of voxels. For BPI, we additionally calculated:

(1) Correct decision rate. The number of correctly classified 'activated,' 'deactivated,' and 'not activated' voxels to its true number (c.f. 'hit rate' in detection theory or 'true positive rate' in frequentist framework).
(2) Incorrect decision rate. The number of voxels incorrectly classified as 'activated,' 'deactivated,' and 'not activated' to the true number of voxels not belonging to 'activated,' 'deactivated,' and 'not activated' categories, respectfully (c.f. 'false alarm rate' in detection theory or 'false positive rate' in frequentist framework);
(3) Low confidence decision rate. The number of 'low confidence' voxels to the total number of voxels.

The code for the simulations is available online[5].

[5]https://github.com/Masharipov/BPI_2021/tree/main/simulations

RESULTS

Results for Contrasts With No Expected Practically Significant Difference

Classical NHST did not show a significant difference between 'Group #1' and 'Group #2' (see **Supplementary Figure 1**). BPI with the 'ROPE-only' decision rule and default ES threshold $\gamma = 1$ *prior* $SD_\theta = 0.190\%$ classified 83.4% of voxels as having 'no difference' in which the null hypothesis was accepted (see **Supplementary Figure 1**). The 'HDI+ROPE' rule classified 76.2% of voxels as having 'no difference.'

Classical NHST did not reveal a significant difference between 'Session #1' and 'Session #2' (see **Supplementary Figure 2**). The *prior* SD_θ was 0.005%. In this case, using the default ES threshold $\gamma = 1$ *prior* SD_θ did not allow the detection of any 'no difference' voxels, because the ROPE was unreasonably narrow. The 'null effect' was detected in all voxels beginning with a $\gamma = 0.013\%$ threshold using the 'ROPE-only' and 'HDI+ROPE' decision rules (see **Supplementary Figure 2**).

In this way, when comparing two 'similar' *independent* samples (two groups of healthy subjects performing the same task), BPI with the default group-level threshold (*one prior* SD_θ) allowed us to correctly label voxels as having 'no difference' for the majority of the voxels of the brain. However, when comparing two 'similar' *dependent* samples (two sessions from the same task), the *one prior* SD_θ threshold became inadequately small.

Therefore, the default *one prior* SD_θ threshold is not suitable when the difference between *dependent* conditions is very small (paired sample test or one-sample test). In such cases, one can use an *a priori* defined ES threshold based on previously reported effect sizes or provide an ES threshold at which most of the voxels can be labeled as having 'no difference,' allowing the critical reader to decide whether this speaks in favor of the absence of differences.

Comparison of Classical Null Hypothesis Significance Testing and Bayesian Parameter Inference Results

Generally, classical NHST with voxel-wise FWE correction and BPI with the 'ROPE-only' decision rule and default group-level ES threshold $\gamma = 1$ *prior* SD_θ revealed similar (de)activation patterns in all considered contrasts (see **Figure 6**, **Table 1**, and **Supplementary Tables 2–10**). The median ES threshold based on $Dice_{max}$ and median default group-level ES threshold across all considered contrasts were close in magnitude to the default subject-level ES threshold $\gamma = 0.1\%$: $\gamma(Dice_{max}) = 0.118\%$ and $\gamma = 1$ *prior* $SD_\theta = 0.142\%$. The median $Dice_{max}$ across all the considered contrasts reached 0.904. At the same time, BPI allowed us to classify 'non-significant' voxels as 'not activated' or 'low confidence.' As it can be clearly seen from **Figure 6**, areas with 'non-activated' voxels surround clusters with 'activated/deactivated' voxels. Both are separated by areas comprising 'low confidence' voxels.

As expected, compared with the 'HDI+ROPE' rule, using the 'ROPE-only' rule slightly increases the number of 'activated/deactivated' and 'not activated' voxels (see **Table 1** and **Supplementary Tables 2–10**). The 'HDI+ROPE' rule labeled more voxels as 'low confidence.'

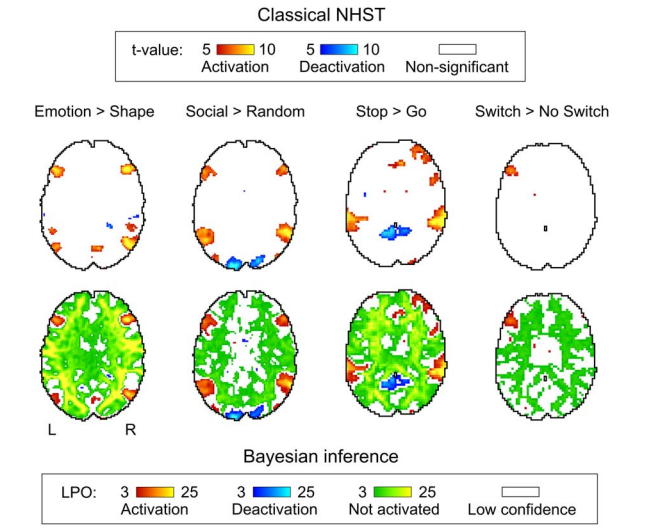


FIGURE 6 | Examples of results obtained with classical NHST and BPI. Four contrasts were chosen for the illustration purposes (two event-related and two block-design tasks). Classical NHST was implemented using voxel-wise FWE correction ($\alpha = 0.05$). BPI was implemented using the 'ROPE-only' decision rule, $P_{thr} = 95\%$ ($LPO > 3$) and $\gamma = 1$ *prior* SD_θ. Axial slice $z = 18$ mm (MNI152 standard space).

Comparison of Bayesian Parameter Inference Results With Different Effect Size Thresholds

Here, we focus on the 'ROPE-only' rule. We first consider the results for the emotional processing task and then consider other tasks. Using the default single-subject ES threshold $\gamma = 0.1\%$ for the emotional processing task ('Emotion > Shape' contrast), 58.8% of all voxels can be classified as 'not activated,' 30.8% as 'low confidence,' and 10.1% as 'activated' (see **Figure 7** and **Supplementary Table 2**). The default group-level ES threshold $\gamma = 1$ *prior* $SD_\theta = 0.135\%$ allowed us to classify 75.0% of all voxels as 'non-activated,' 17.5% as 'low confidence,' and 7.4% as 'activated' (see **Figure 7** and **Supplementary Table 2**). Both types of thresholds were comparable to those of classical NHST for the detection of 'activated' voxels. The maximum overlap between 'activations' patterns revealed by classical NHST and BPI was observed at $\gamma(Dice_{max}) = 0.116\%$ (see **Figure 8** and **Table 1**).

For the '2-back > 0-back,' 'Left Finger > baseline,' 'Right Finger > baseline,' and 'Social > Random' contrasts, the three ES thresholds that were considered—0.1%, *one prior* SD_θ, $\gamma(Dice_{max})$—produced similar results (see **Table 1** and **Supplementary Tables 3**, **5**, **7**). For the event-related stop-signal task ('Correct Stop > baseline' and 'Correct Stop > Go' contrasts), *one prior* SD_θ and $\gamma(Dice_{max})$ were close in terms of their values but smaller than 0.1% (see **Table 1**). Block designs tend to evoke higher BOLD PSC than event-related designs; therefore,

TABLE 1 | Maximum Dice coefficient and corresponding effect size thresholds for each task.

Contrast, θ	*Prior* SD_θ, %	'ROPE-only' decision rule		'HDI+ROPE' decision rule	
		$\gamma(Dice_{max})$, %	$Dice_{max}$	$\gamma(Dice_{max})$, %	Dice
Emotion processing					
Emotion > Shape	0.135	0.116	0.904	0.104	0.912
Working memory					
2-back > baseline	0.325	0.136	0.925	0.125	0.932
2-back > 0-back	0.089	0.095	0.891	0.089	0.903
Language					
Story > Math	0.255	0.119	0.896	0.108	0.904
Motor					
Left finger > baseline	0.149	0.148	0.897	0.135	0.907
Right finger > baseline	0.171	0.160	0.886	0.144	0.897
Tongue > baseline	0.268	0.205	0.904	0.181	0.913
Gambling					
Reward > baseline	0.254	0.132	0.917	0.122	0.924
Loss > baseline	0.249	0.134	0.918	0.118	0.925
Reward > Loss	0.032	0.044	0.894	0.037	0.886
Social cognition					
Social > baseline	0.325	0.139	0.939	0.124	0.944
Social > Random	0.104	0.114	0.896	0.104	0.907
Relational processing					
Relational > baseline	0.390	0.154	0.935	0.143	0.940
Relational > Match	0.051	0.073	0.892	0.066	0.894
Stop-signal task					
Correct Stop > baseline	0.069	0.066	0.895	0.061	0.906
Correct Stop > Go	0.064	0.052	0.906	0.047	0.917
Task-switching					
Switch > baseline	0.133	0.075	0.907	0.067	0.916
Switch > No switch	0.030	0.037	0.924	0.033	0.925
Summary					
Median	0.142	0.118	0.904	0.106	0.913

a lower *prior* SD_θ should be expected for event-related designs and higher *prior* SD_θ for block designs. Within a single design, in contrasts such as 'task-condition > baseline,' higher BOLD PSC and *prior* SD_θ would be expected than in contrasts in which the experimental conditions are compared directly. For example, the contrast '2-back > baseline' has *prior* $SD_\theta = 0.325\%$ and contrast '2-back > 0-back' has *prior* $SD_\theta = 0.089\%$.

As previously noted, some contrasts did not elicit robust activations: 'Reward > Loss,' 'Relational > Match,' (Barch et al., 2013) and 'Switch > No switch' (Gorgolewski et al., 2017). The corresponding $\gamma(Dice_{max})$ thresholds were 0.044, 0.073, and 0.037% (see **Table 1** and **Supplementary Tables 6**, **8**, **10**). The *prior* SD_θ were 0.032, 0.051, and 0.030%. Correspondingly, BPI with the $\gamma = 1$ *prior* SD_θ threshold classified 0, 18.4, and 42.2% of voxels as 'not activated.' This demonstrates that when we compare conditions with similar neural activity and minor differences, it becomes more difficult to separate 'activations/deactivations' from the 'null effects' using the $\gamma = 1$ *prior* SD_θ threshold.

Typical Effect Sizes in Functional Magnetic Resonance Imaging Studies

A complete list of effect sizes (BOLD PSC and Cohen's d) estimated for different tasks and *a priori* defined ROIs is presented in the **Supplementary Materials** (**Supplementary Tables 11**–**19**). Here, we focus only on the BOLD PSC. The violin plots for some of these are shown in **Figure 9**.

For example, the median BOLD PSC in the left amygdala ROI, one of the key brain areas for emotional processing, was 0.263%, which is approximately twice as large as *one prior* SD_θ (see **Figure 7**). Thus, using this PSC as the ES threshold in future studies may cause the ROPE to become too wide compared to the effect sizes typical for tasks with such designs. Therefore, such a threshold can be used to detect large and highly localized effects. However, it may fail to detect small but widely distributed effects previously described for HCP data (Cremers et al., 2017).

In general, median PSCs within ROIs were up to 1% for block designs and 0.5% for event-related designs. The maximum PSCs reached 2.5% and were usually observed in the primary visual cortex (V1) for visual tasks comparing

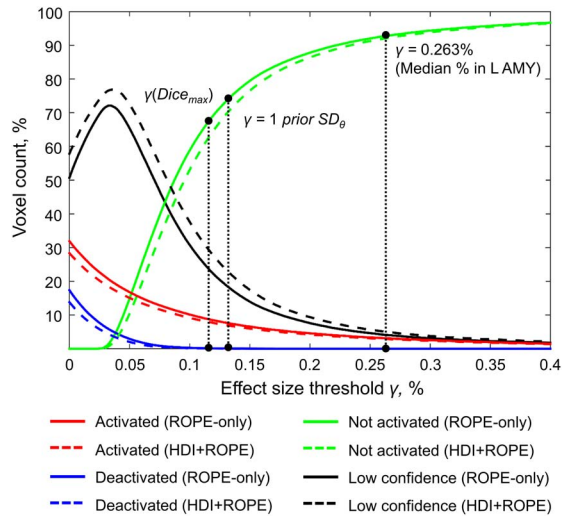


FIGURE 7 | Number of voxels classified into the four categories depending on the ES threshold γ. The results for the emotion processing task ('Emotion>Shape' contrast) are presented for illustration. L AMY, left amygdala.

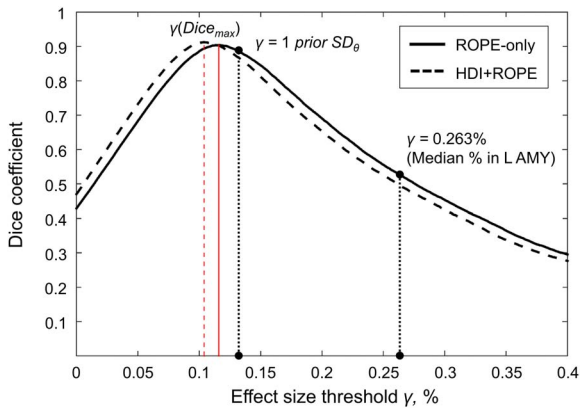


FIGURE 8 | Dependence of the Dice coefficient on the ES threshold γ. Results for the emotion processing task ('Emotion>Shape' contrast). The red lines denote $\gamma(Dice_{max})$. L AMY, left amygdala.

experimental conditions with baseline activity. For 'moderate' physiological effects, PSC varied in the range 0.1−0.2%, for example, for the '2-back > 0-back' contrast, the median PSC in the right dorsolateral prefrontal cortex (R DLPFC in **Figure 9**) was 0.137%. Likewise, for the 'Social > Random' contrast, the right inferior parietal lobule (R IPL) median PSC was 0.137%, for the 'Correct Stop > Go,' the right inferior frontal gyrus/frontal operculum (R IFG/FO) median

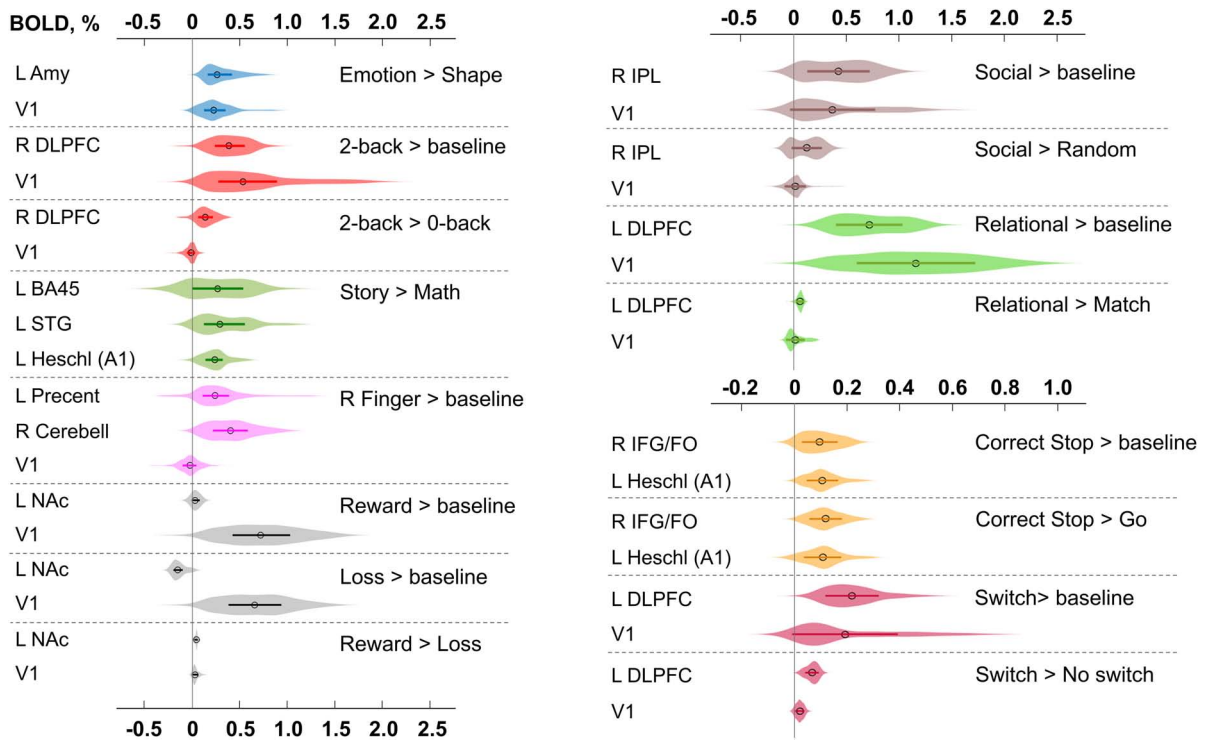


FIGURE 9 | Typical BOLD PSC in fMRI studies. The box plots inside the violins represent the first and third quartile, and the black circles represent median values. Contrasts from the same task are indicated in one color. L/R, left/right; AMY, amygdala; V1, primary visual cortex; DLPFC, dorsolateral prefrontal cortex; BA, Brodmann area; STG, superior temporal gyrus; A1, primary auditory cortex; NAc, nucleus accumbens; IPL, inferior parietal lobule; IFG/FO, inferior frontal gyrus/frontal operculum.

PSC was 0.120%. For more 'strong' physiological effects, the PSC was in the range 0.2−0.3%, for example, for the 'Emotion > Shape' contrast, the median PSC in the left amygdala was 0.263%, and for the 'Story > Math' contrast, the median PSC in the left Brodmann area 45 (Broca's area) was 0.269%. For the motor activity, for example the 'Right Finger > baseline' contrast, the median PSC in the left precentral gyrus was 0.239%, in the left postcentral gyrus was 0.362%, in the left putamen was 0.290%, and in the right cerebellum was 0.401%. For the contrasts that did not elicit robust activations (Barch et al., 2013), the PSC was approximately 0.05–0.1%; for example, for the 'Reward > Loss' contrast, the median PSC in the left nucleus accumbens was 0.043%, and for the 'Relational > Match' contrast, the median PSC in the left dorsolateral prefrontal cortex was 0.062%.

Region of Practical Equivalence Maps

We considered BPI with two consecutive thresholding steps: (1) calculate the *LPOs* (or PPMs) with a selected ES threshold γ, (2) apply the posterior probability threshold p_{th} = 95% or consider the overlap between the 95% HDI and ROPE. We can *reverse the thresholding sequence* and calculate *the ROPE maps*.

For the 'activated/deactivated' voxels, the ROPE map contains the maximum ES thresholds that allow voxels to be classified as 'activated/deactivated' based on the 'ROPE-only' or 'HDI+ROPE' decision rules. For the 'not activated' voxels, the map contains the minimum effect size thresholds that allow voxels to be classified as 'not activated.'

The procedure for calculating the ROPE map can be performed as follows. Let us consider a gradual increase in the ROPE radius (i.e., the half-width of ROPE or the ES threshold γ) from zero to the maximum effect size in observed volume. (1) For voxels in which PSC is close to zero, at a certain ROPE radius, the posterior probability of finding the effect within the ROPE becomes higher than 95%. This width is indicated on the ROPE map for 'not activated' voxels. (2) For voxels in which the PSC deviates from zero, at a certain ROPE radius, the posterior probability of finding the effect outside the ROPE becomes lower than 95%. This width is indicated on the ROPE map for 'activated/deactivated' voxels. The same maps can be calculated for the 'HDI+ROPE' decision rule.

Examples of the ROPE maps are shown in **Figure 10**. From our point of view, ROPE maps, as well as unstandardized effect size (PSC) maps, may facilitate an intuitive understanding of the spatial distribution of a physiological effect under investigation (Chen et al., 2017). They can also be a valuable addition to standard PPMs, allowing researchers to flexibly choose the ES threshold based on expected effect size for specific experimental conditions, ROIs and MR acquisition parameters. The default ES thresholds may be more conservative to brain areas near air–tissue interfaces due to signal dropout. The researcher may choose a lower ES threshold to increase sensitivity to these brain areas.

Effects of Spatial Smoothing on Classical Null Hypothesis Significance Testing and Bayesian Parameter Inference

Two main effects of spatial smoothing were identified. Firstly, higher spatial smoothing increased the number of both 'activated/deactivated' and 'not activated' voxels classified by BPI, reducing the number of 'low confidence' voxels. Secondly, higher smoothing blurred the spatial localisation of local maxima of t-maps and PPMs (*LPO*-maps) to a different extent. Consider, for example, the emotion processing task ('Emotion > Shape' contrast). The broadening of two peaks in the left and right amygdala was more noticeable on the t-map than on the PPM (see **Figure 11**).

Smoothing was previously shown to have a nonlinear effect on the voxel variances and thus to affect more t-maps than β value maps, sometimes leading to counterintuitive artifacts (Reimold et al., 2005). This is especially noticeable at the border between two different tissues or between the two narrow peaks of the local maxima. If the peak is localized close to white matter voxels with low variability, then smoothing can shift the peak to the white matter. If low-variance white matter voxels separate two close peaks, then after smoothing, they may serve as a 'bridge' between the two peaks. To avoid this problem, Reimold et al. (2005) recommended using masked β value maps. In the present study, we suggest that PPMs based on BOLD PSC thresholding can mitigate this problem. Importantly, smoothing artifacts can also arise on Cohen's d maps. Therefore, PPMs based on PSC thresholding may be preferable to PPMs based on Cohen's d thresholding.

Sample Size Dependencies for Classical Null Hypothesis Significance Testing and Bayesian Parameter Inference

An enlargement of the sample size led to an increase in the number of 'activated' and 'not activated' voxels, and a decrease in the number of 'low confidence' voxels. This is due to a decrease in the posterior variance. The curve of the 'activated' voxels rose much slower than that of the 'not activated' voxels. For the emotion processing task ('Emotion > Shape' contrast, block-design, two sessions, 352 scans), the largest gain in the number of 'activated' and 'not activated' voxels can be noted from 20 to 30 subjects (see **Figure 12A**). With a sample size of $N > 30$, the number of 'activated' and 'not activated' voxels increased less steeply. The 'not activated' and 'low confidence' voxels curves intersected at $N = 30$ subjects. After the intersection point, the graphs reached a plateau.

Considering only half of the emotional processing task data (one session, 176 scans), the intersection point shifted from $N = 30$ to $N = 60$ (see **Figure 12B**). For the event-related task ('Correct Stop > Go' contrast, the stop-signal task, 184 scans), all considered dependencies had the same features as for the block-design task, and the point of intersection was at $N = 60$ subjects (see **Figure 12C**). For the fixed ES threshold, the moment at which the graphs reach a plateau depends on task design, data quality and the amount of data at the subject level, that is, on

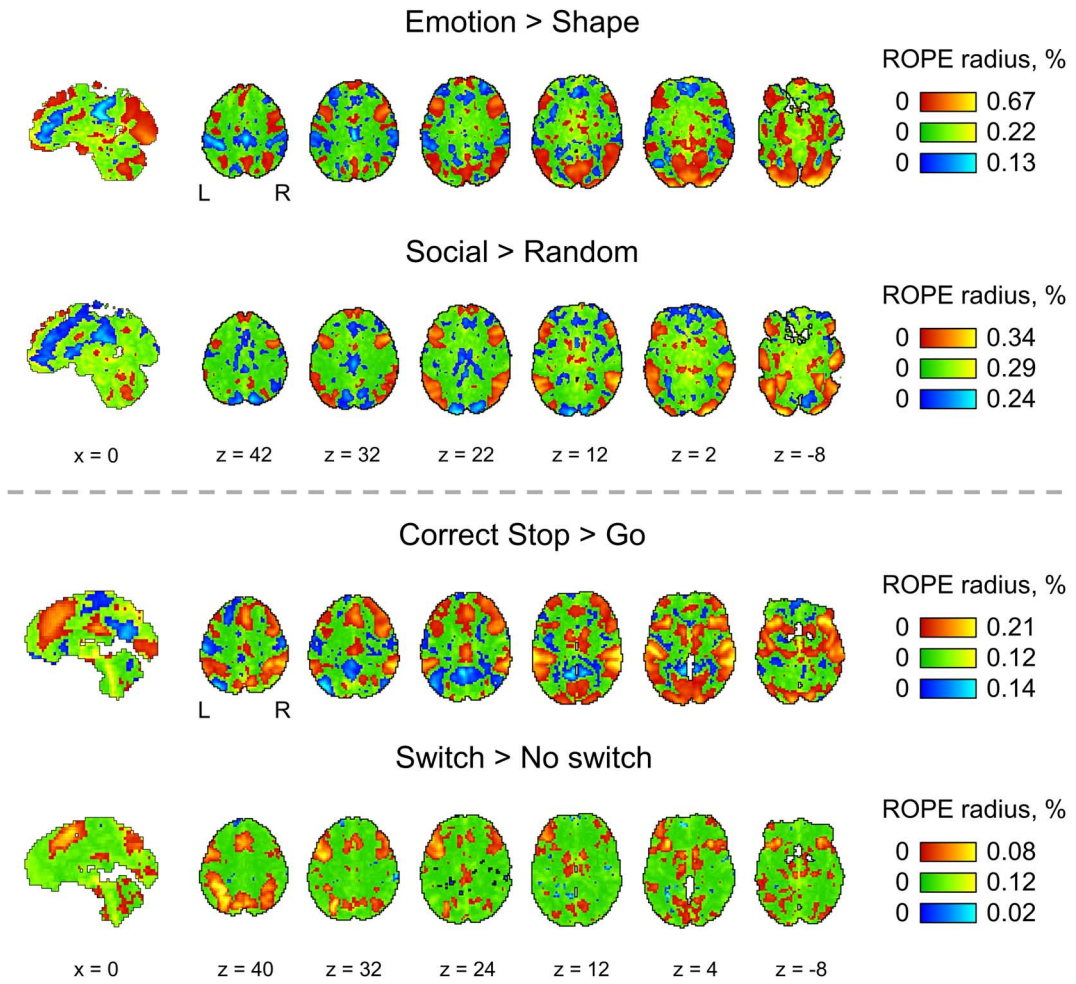


FIGURE 10 | The ROPE maps. Four contrasts were chosen for the illustration purposes (two event-related and two block-design tasks). The ROPE maps are presented using different colors for the 'activated,' 'deactivated,' and 'not activated' voxels. The green bars represent the minimum ROPE radii at which voxels with a PSC close to zero can be classified as 'not activated' based on the 'ROPE-only' decision rule. The red and blue bars represent the maximum ROPE radii at which voxels of which the PSC deviates from zero can be classified as 'activated' and 'deactivated,' respectively.

the number of scans, blocks, and events. The task designs from the HCP and UCLA datasets have relatively short durations (for example, the stop-signal task has approximately 15 'Correct Stop' trials per subject). Studies with a shorter scanning time generally require a larger sample size to enable inferences to be made with confidence. Lowering the ES threshold would also require larger sample size to reach a plateau.

Classical NHST with the voxel-wise FWE correction showed a steady linear increase in the number of 'activated' voxels with increasing sample size (see **Figure 13**). With a further increase in the sample size, the number of statistically significant voxels revealed by classical NHST is expected to approach 100% (see, for example, Gonzalez-Castillo et al., 2012; Smith and Nichols, 2018). In contrast, the BPI with the $\gamma = 1$ *prior* SD_θ threshold demonstrated hyperbolic dependencies. We observed a steeper increase at small and moderate sample sizes ($N = 15-60$). The curve of the 'activated' voxels flattened at large sample sizes ($N > 80$). BPI offers protection against the detection of 'trivial' effects that can appear as a result of an increased sample size if classical NHST with the point-null hypothesis is used (Friston et al., 2002a; Friston, 2012; Chen et al., 2017). This is achieved by the ES threshold γ, which eliminates physiologically (practically) negligible effects. **Figure 13** presents an illustration of the Jeffreys-Lindley paradox, that is, the discrepancy between results obtained using classical and Bayesian inference, which is usually manifested at higher sample sizes (Jeffreys, 1939/1948; Lindley, 1957; Friston, 2012).

Normality Check

For the block-design task ('Emotion > Shape' contrast), the number of significantly non-Gaussian voxels was 17% with $\alpha_{uncorr} = 0.001$ and 2% with $\alpha_{Bonf} = 0.05$. The median kurtosis and skewness across voxels was $Ku = 3.77$ and $Sk = 0.05$. For the event-related task ('Correct Stop > Go' contrast), the number of significantly non-Gaussian voxels was 19% with $\alpha_{uncorr} = 0.001$ and 4% with

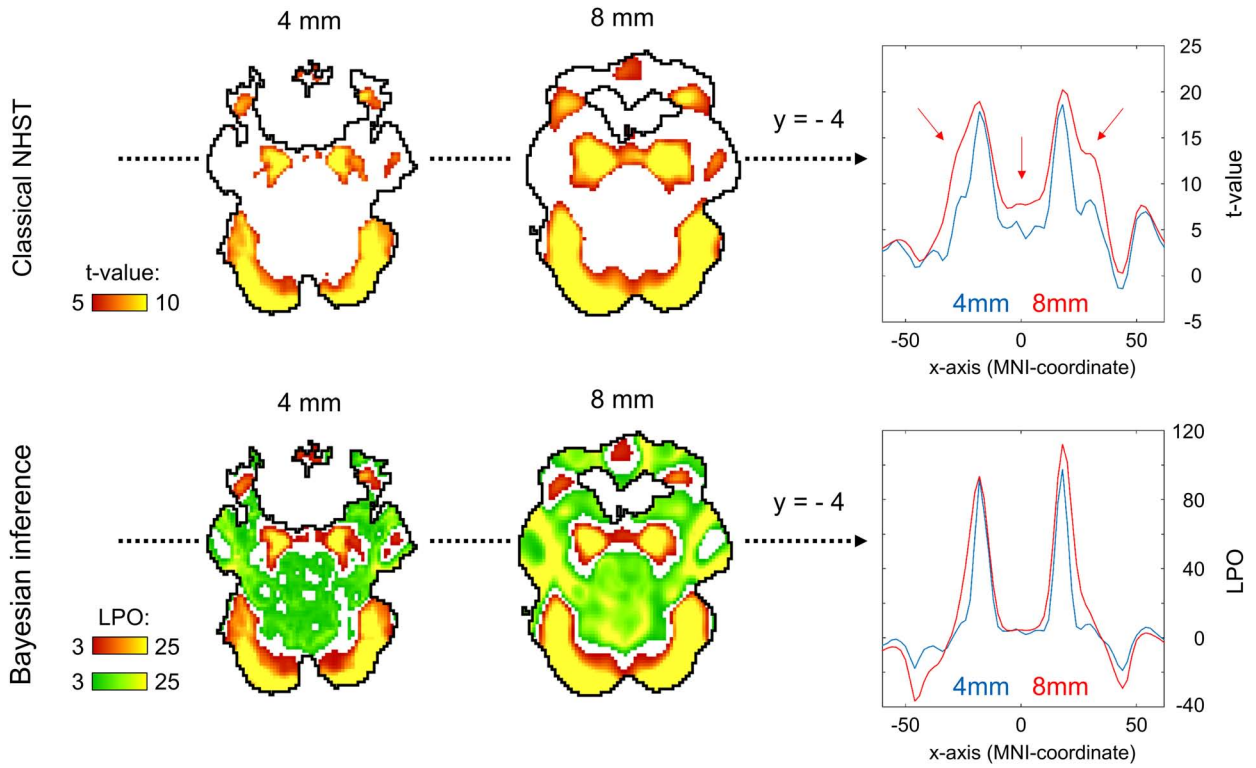


FIGURE 11 | Influence of spatial smoothing on classical NHST and BPI: results for the emotion processing task ('Emotion > Shape' contrast). Classical NHST was implemented using voxel-wise FWE correction ($\alpha = 0.05$). BPI was implemented using the 'ROPE-only' decision rule, $P_{thr} = 95\%$ ($LPO > 3$) and $\gamma = 1$ *prior* $SD_\theta = 0.135\%$. Axial slice $z = -14$ mm (MNI152 standard space). Slice images have different outlines due to spatial smoothing (higher spatial smoothing increases the size of implicit masks for single subjects and group of subjects). In the panels on the right, 1-D images are presented for t-values and *LPOs* along the x-axis for $y = -4$ mm. The red arrows indicate a noticeable broadening of two peaks of local maxima (left and right amygdala) at higher smoothing.

$\alpha_{Bonf} = 0.05$. The median kurtosis and skewness across voxels was $Ku = 3.77$ and $Sk = 0.05$. In general, the data are consistent with the normality assumption, though some voxels violate it.

Simulations

The simulations results reproduced the results obtained from the empirical data (see **Figure 14** for an overview of the simulations). Further, they allowed us to demonstrate how various factors affect BPI performance with the known ground truth.

Dependence of the Number of 'Activated' Voxels on the Sample Size

The number of 'activated' voxels revealed by BPI with the $\gamma = 1$ *prior* SD_θ threshold approaches the true number of practically significant voxels and stops increasing (see **Figure 15**). Classical NHST shows further increase of 'activated' voxels with the sample size increase, as it considers only statistical significance. This is more evident for low and medium noise cases ($SD = 0.2, 0.3\%$). For the high noise case ($SD = 0.4\%$), the sample size should be larger than $N = 500$ for the discrepancy between NHST and BPI results to become evident.

Dependence of the Correct and Low Confidence Decision Rates on the Sample Size

For the weak effect size ($\theta = 0.1\%$), the BPI with the $\gamma = 1$ *prior* SD_θ threshold is more sensitive for 'activated' than for 'not activated' voxels (see **Figure 16**). This is because $\gamma = 1$ *prior* SD_θ threshold is smaller for the weak effect size. For the moderate and strong effects ($\theta = 0.2, 0.3\%$), this difference in sensitivity become less evident. The low confidence decisions are prevalent in the 'weak effect plus high noise' case. It becomes more challenging to distinguish between 'activated' and 'not activated' voxels when the data are noisy, and the PSC in the 'activated' voxels is close to the PSC in 'trivial' voxels. For the intermediate case (moderate effect plus medium noise), the correct decision rates for 'activated' and 'not activated' voxels reached 80% at the sample sizes $N = 80$ and $N = 150$, correspondingly. For larger effect sizes and lower noise, a smaller sample size will be required to achieve the correct decision rate of 80% (and vice versa). The 'ROPE-only' decision rule is more sensitive to both 'activated' and 'not activated' voxels than the 'HDI+ROPE' decision rule.

Robustness of Bayesian Parameter Inference to Violations of the Normality Assumption

Non-normal distributions with positive and negative skewness increase incorrect decision rates for 'deactivated' and 'activated'

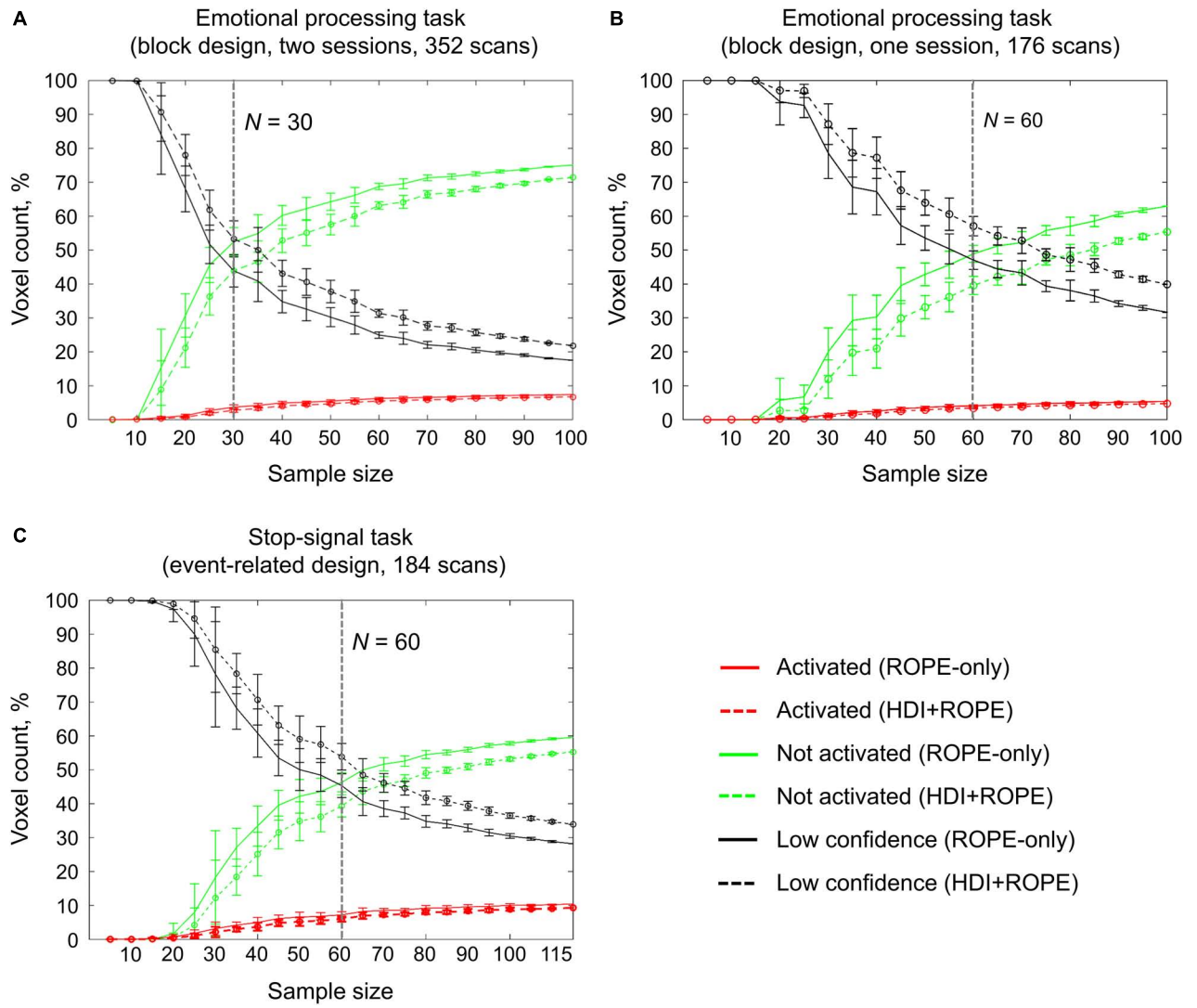


FIGURE 12 | Dependencies of the number of 'activated,' 'not activated,' and 'low confidence' voxels on the sample size. BPI was implemented using $\gamma = 1$ *prior* SD_θ. **(A)** The emotional processing task ('Emotion > Shape' contrast, two sessions). **(B)** The emotional processing task ('Emotion > Shape' contrast, one session). **(C)** The stop-signal task ('Correct Stop > Go' contrast). The error bars represent the mean and standard deviation across ten random groups.

voxels, correspondingly (**Figure 17**). Application the 'ROPE-only' decision rule results in higher incorrect decision rates than the 'HDI+ROPE' decision rule. However, even in the worst case (weak effect plus high noise), the incorrect decision rates for BPI with the $\gamma = 1$ *prior* SD_θ threshold did not exceed 5%. This result shows that BPI is robust to violations of the normality assumption. The 'ROPE-only' rule may be preferable to the 'HDI+ROPE' rule, as both rules protect against incorrect decisions, but the 'ROPE-only' rule is more sensitive to the true effects using $\gamma = 1$ *prior* SD_θ threshold.

Dependence of the Correct and Incorrect Decision Rates on the Effect Size Threshold

The optimal ES threshold should provide high sensitivity to both 'activated' and 'not activated' voxels (e.g., higher than 80%) while protecting against incorrect decisions (e.g., lower than 5%). The range of ES thresholds that meets these criteria decreases for lower true effects and higher noise (see **Figure 18**). At the sample size $N = 200$, the default $\gamma = 1$ *prior* SD_θ threshold falled in the range of optimal ES thresholds in the majority of the cases. For the weak effect plus high noise case, one should choose between high sensitivity to 'activated' or 'not activated' voxels. In this scenario, to achieve high sensitivity to both types of voxels, it is necessary to obtain a very large sample size ($N > 500$). In all considered cases, the default ES threshold provided approximately equal correct decision rates for 'activated' and 'not activated' voxels and protected against incorrect decisions. This result confirmed that the default IS threshold is optimal in most scenarios, except for the scenario with low effect and high noise level.

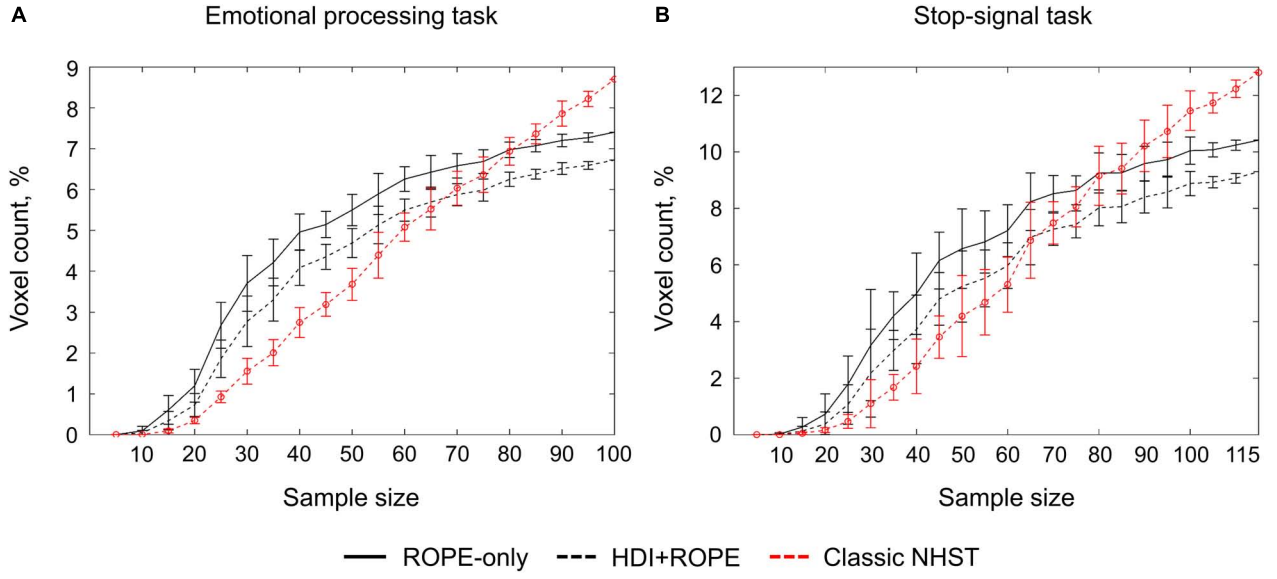


FIGURE 13 | Dependencies of the number of 'activated' voxels on the sample size. Classical NHST was implemented using FWE correction ($\alpha = 0.05$). BPI was implemented using $\gamma = 1$ *prior* SD_θ. **(A)** The emotional processing task (block design, 'Emotion > Shape' contrast). **(B)** The stop-signal task (event-related design, 'Correct Stop > Go' contrast). The error bars represent the mean and standard deviation across ten random groups.

Example of Practical Application of Bayesian Parameter Inference

In contrast to classical NHST, Bayesian inference allows us to:

(1) Provide evidence that there is no practically meaningful BOLD signal change in the brain area when comparing the two task conditions.
(2) Establish double dissociations; that is to state that one area responds to A *but not* B condition and another responds to B *but not* A condition (Friston et al., 2002a).
(3) Provide evidence for practically equivalent engagement of one area under different experimental conditions in terms of local brain activity.
(4) Provide evidence for the absence of a practically meaningful difference in BOLD signals between groups of subjects or repeated measures.

To illustrate a possible application of Bayesian inference in research practice, we used a working memory task. Let us consider an overlap between the '2-back > baseline' and '0-back > baseline' contrasts (see **Figure 19**, purple areas). We cannot claim that brain areas revealed by this conjunction analysis were equally engaged in the '2-back' and '0-back' conditions. To provide evidence for this notion, we can use BPI and attempt to identify voxels with a practically equivalent BOLD signal in the '2-back' and '0-back' conditions (see **Figure 19**, green areas). Overlap between the '2-back > baseline' and '0-back > baseline' and the '2-back = 0-back' effects was found in several brain areas: visual cortex (V1, V2, V3), frontal eye field (FEF), superior eye field (SEF), parietal eye field (PEF, or posterior parietal cortex), lateral geniculate nucleus (LGN) and left primary motor cortex (M1) (see **Figure 19**, white areas). This result can be easily explained by the fact that both experimental conditions require the subject to analyze perceptually similar visual stimuli and push response buttons with the right hand, which should not depend much on the working memory load. At the same time, it does not follow directly from simple conjunction analysis.

DISCUSSION

Over-reliance on classical NHST promotes publication bias toward statistically significant findings. However, the null result can be just as valuable and exciting as the statistically significant result. Furthermore, not every statistically significant result has a practical significance. In recent years, statistical practice has seen a gradual shift from point-null hypothesis testing to interval-null hypothesis testing and interval estimation, as well as from frequentist to Bayesian approaches. Frequentist and Bayesian interval-based approaches allow us to assess the 'null effects' and thus overcome prejudice against the null hypothesis. While both approaches may lead to similar results (if specially calibrated to get it), we discussed conceptual and practical reasons for preferring the Bayesian approach. One of the main conceptual difficulties of the frequentist approach is that it is based on the probabilistic 'proof by contradiction,' which results in the 'inverse probability' fallacy: that is a widespread misinterpretation of *p*-values and confidence intervals as posterior probabilities and credible intervals. Although the Bayesian approach does not automatically guarantee correct interpretations, it can be more intuitive and straightforward than the frequentist approach (particularly, Bayesian inference based on the posterior probability distributions of the parameters or BPI).

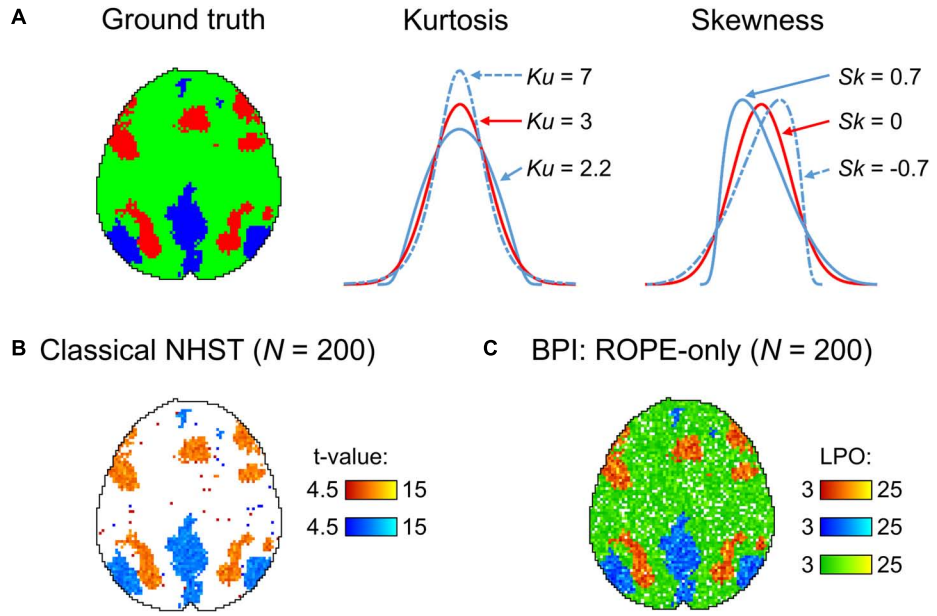


FIGURE 14 | Simulations overview. **(A)** Ground truth axial slice z = 36 mm (MNI152 standard space). 'Activated' and 'deactivated' voxels are marked in red and blue colors, respectfully. 'Trivial' voxels that should be classified as 'not activated' (practically equivalent to the null value) are marked in green. Data were drawn from the normal (Ku = 3, Sk = 0, the red line) and non-normal distributions. **(B)** Classical NHST results for N = 200 images, moderate effect and medium noise (θ = 0.2%, SD = 0.3%), obtained using voxel-wise FWE correction (α = 0.05). **(C)** BPI results for N = 200 images, moderate effect and medium noise (θ = 0.2%, SD = 0.3%), obtained using the 'ROPE-only' decision rule, P_{thr} = 95% ($LPO > 3$) and γ = 1 *prior* SD_θ.

At the same time, from the frequentist point of view, the main conceptual disadvantage of the Bayesian approach is the need to specify our prior beliefs about the model parameters. Sometimes it is argued that we do not want our result to depend on a subjective prior decision. However, in the frequentist framework, we also make prior assumptions when subjectively choosing a model or ignoring the prior distributions of model parameters (implicitly use 'flat' prior). From this point of view, the explicit choice of the prior may be rather an advantage. We can choose prior from theoretical arguments (e.g., biophysical or anatomical priors) or derive prior from the hierarchically organized data (empirical Bayes approach). In this way, we limit the subjectivity of the choice of the prior.

Another potential obstacle to using Bayesian statistics is its computational complexity. Integrals in Bayes' rule can be solved analytically only for relatively simple models. In other cases, numerical integration approaches should be used to calculate the posterior probability, which are particularly time-consuming, especially when considering thousands of voxels. Alternatively, one can use computationally efficient analytical approximations to the posterior distributions, which, however, can be less accurate for high-dimensional parameter spaces (multivariate analysis).

Despite profound development of Bayesian techniques, to date, the 'null effect' assessment is uncommon in neuroimaging field and, in particular, in fMRI studies. One of the possible reasons for this may be the lack of tools available to the end-user. To facilitate the 'null effect' assessment for fMRI practitioners, we developed SPM12 based toolbox for group-level Bayesian inference[4]. We evaluated the BPI approach on empirical and simulated data and discussed its possible application in fMRI studies.

Bayesian parameter inference allows us to simultaneously find 'activated/deactivated,' 'not activated,' and 'low confidence' voxels using a single decision rule. The 'not activated' decision means that the effect is practically non-significant and can be considered equivalent to the null for practical purposes. The 'low confidence' decision means we need more data to make a confident inference, that is, we need to increase the scanning time, sample size, data quality or revisit the task design. The use of parametrical empirical Bayes with the 'global shrinkage' prior enables us to check the results as the sample size increases and allows us to decide whether to stop the experiment if the obtained data are sufficient to make a confident inference. All the above features are absent from the classical NHST framework, limited to the point-null hypothesis with a pre-determined stopping rule.

An important advantage of Bayesian inference is that we can use graphs such as those shown in **Figure 12** to determine when the obtained data are sufficient to make a confident inference. We can plot such graphs for the whole brain or for *a priori* defined ROIs. When the curves reach a plateau, the data collection can be stopped. If the brain area can be labeled as either 'activated/deactivated' or 'not activated' at a relatively small sample size, it will be still so at larger sample sizes. If the brain area can be labeled as 'low confidence,' we must increase the sample size to make a confident inference. At a certain sample size, it could possibly be labeled as either 'activated/deactivated' or 'not activated.' In the worst case, we can

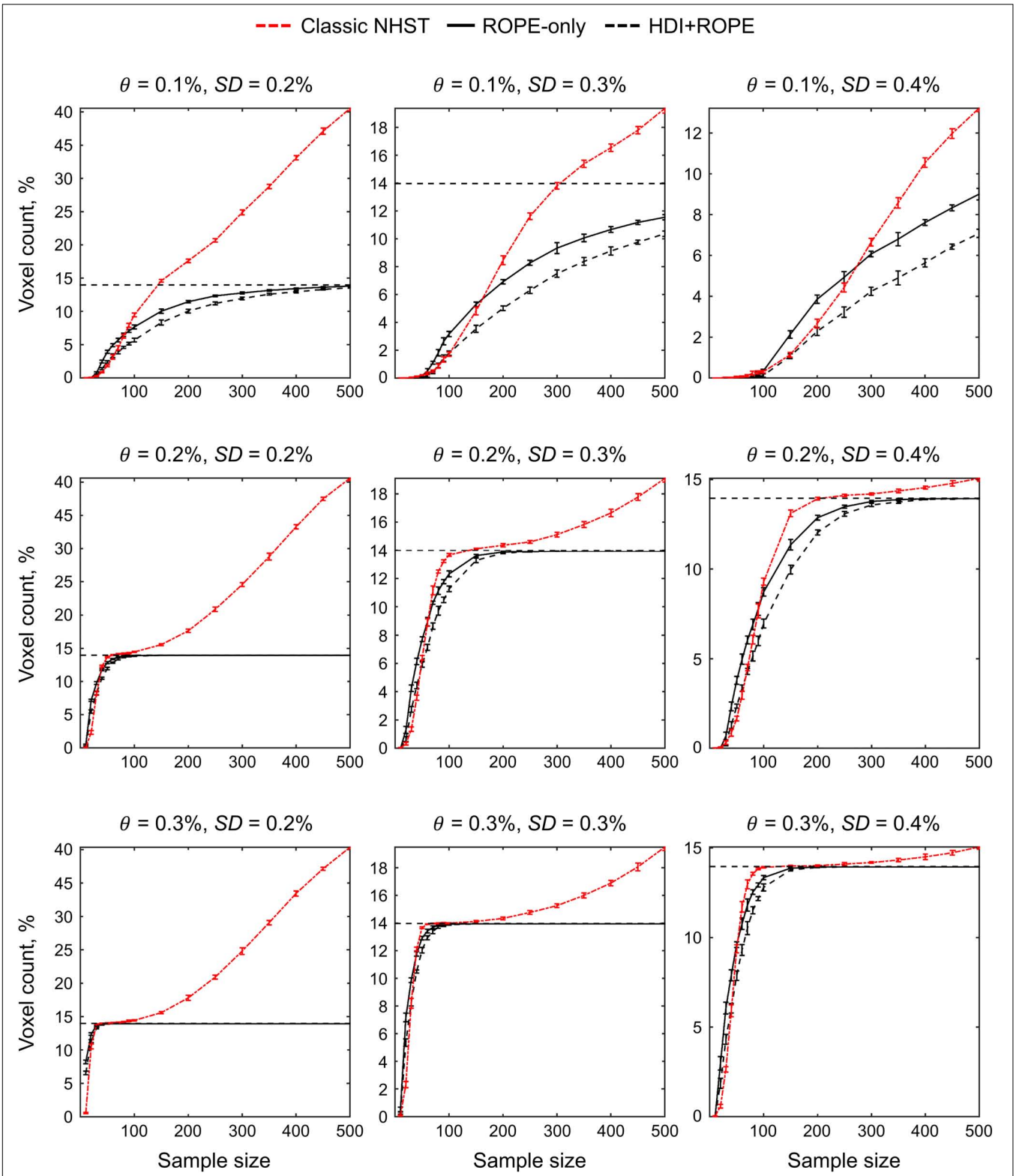


FIGURE 15 | Simulations results for the dependencies of the number of 'activated' voxels on the sample size. Data were drawn from normal distributions with different mean effect θ and noise *SD*. Classical NHST was implemented using FWE correction (α = 0.05). BPI was implemented using γ = 1 *prior* SD_θ. The error bars represent the mean and standard deviation across ten random groups. Horizontal lines indicate the true number of practically significant voxels.

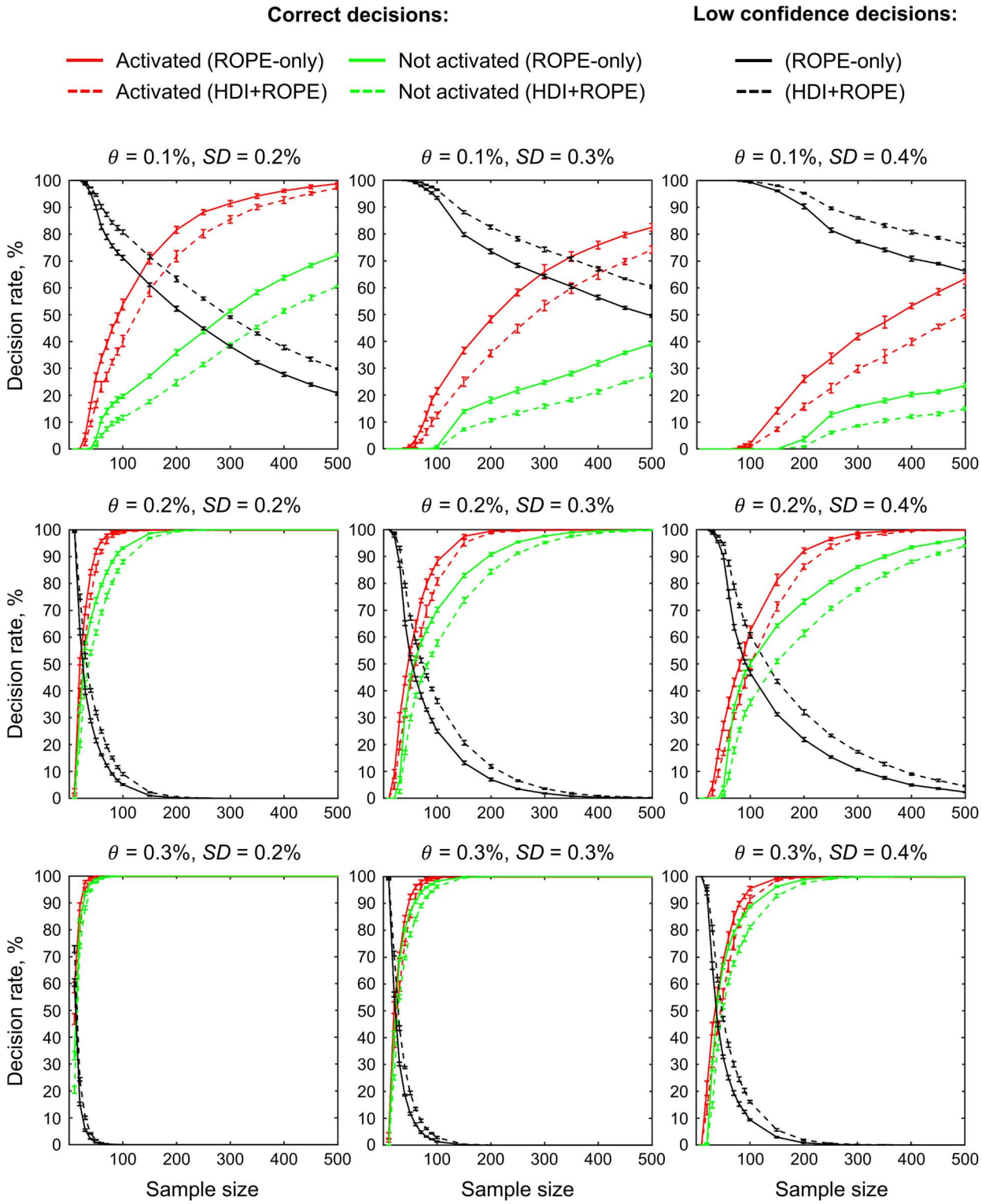


FIGURE 16 | Simulations results for the dependencies of the correct and low confidence decision rates on the sample size. Data were drawn from normal distributions with different mean effect θ and noise *SD*. BPI was implemented using $\gamma = 1$ *prior* SD_θ. The plots for 'deactivated' voxels closely follow the plots for 'activated' voxels and have therefore been omitted for visualization purposes. The error bars represent the mean and standard deviation across ten random groups.

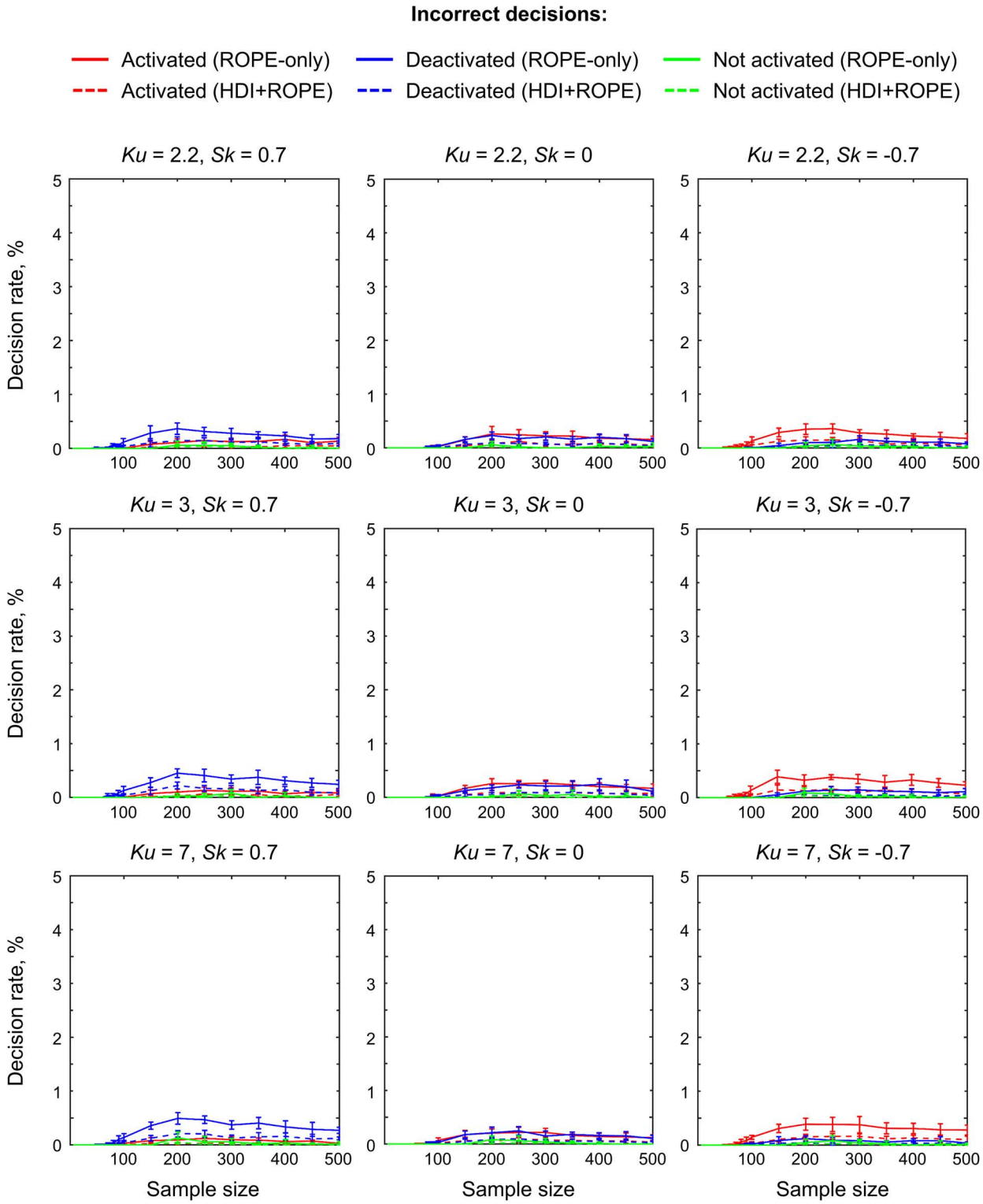


FIGURE 17 | Simulations results for the dependencies of the incorrect decision rate on the sample size. Data were drawn from normal (*Ku* = 3, *Sk* = 0) and non-normal distributions with weak effect and high noise (θ = 0.1%, *SD* = 0.4%). BPI was implemented using γ = 1 *prior* SD_θ. The error bars represent the mean and standard deviation across ten random groups.

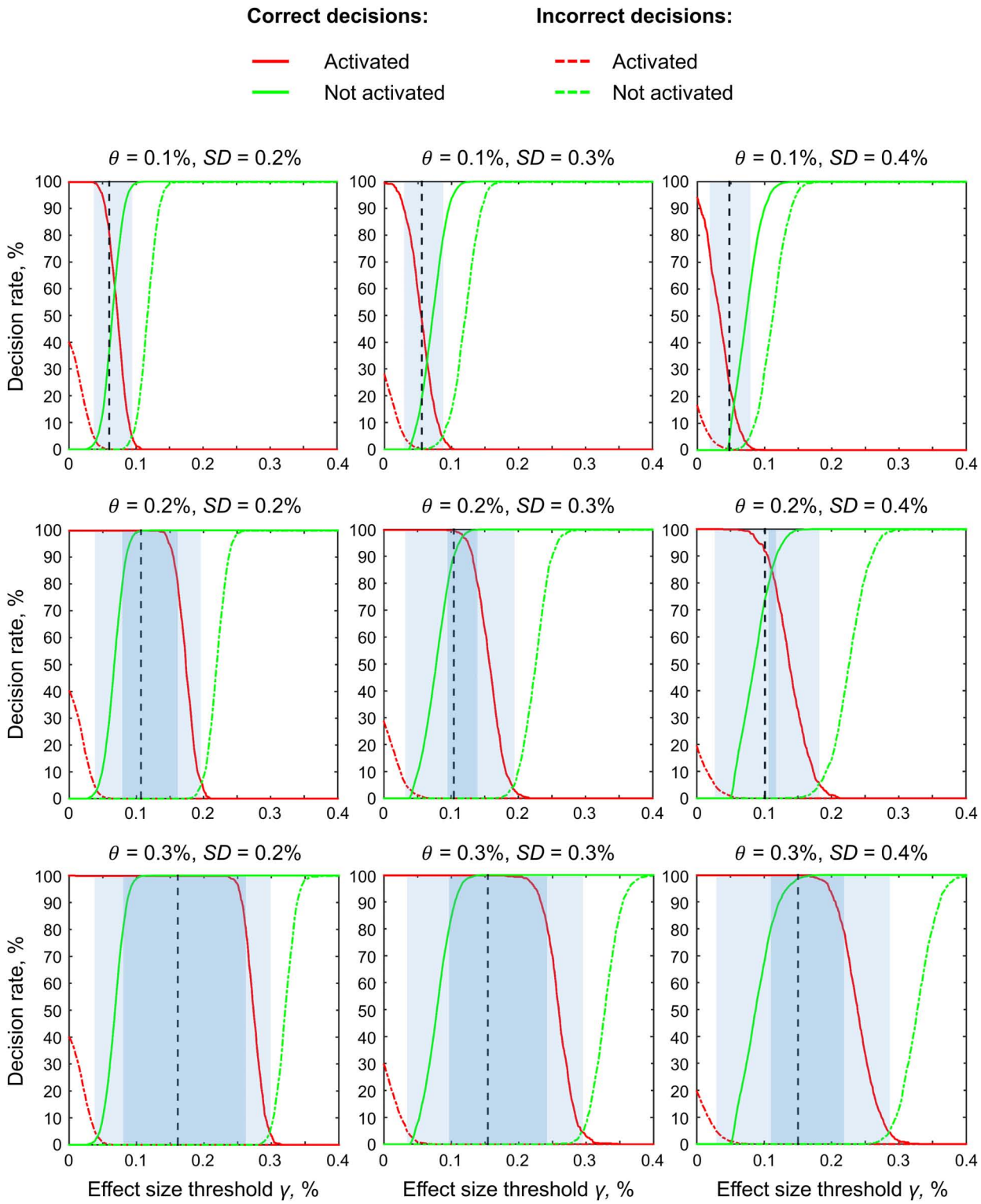


FIGURE 18 | Simulations results for the dependencies of the correct and incorrect decision rates on the ES threshold γ. Data were drawn from normal distributions with different mean effect θ and noise *SD*. Sample size *N* = 200 images, results for one random group. The plots for 'deactivated' voxels closely follow the plots for 'activated' voxels and have therefore been omitted for visualization purposes. Vertical lines indicate the default ES threshold γ = 1 *prior* SD_θ. The light blue areas indicate ES thresholds at which the incorrect decision rates do not exceed 5% for both 'activated' and 'not activated' voxels. The dark blue areas indicate ES thresholds at which the correct decision rates exceed 80% for both 'activated' and 'not activated' voxels.

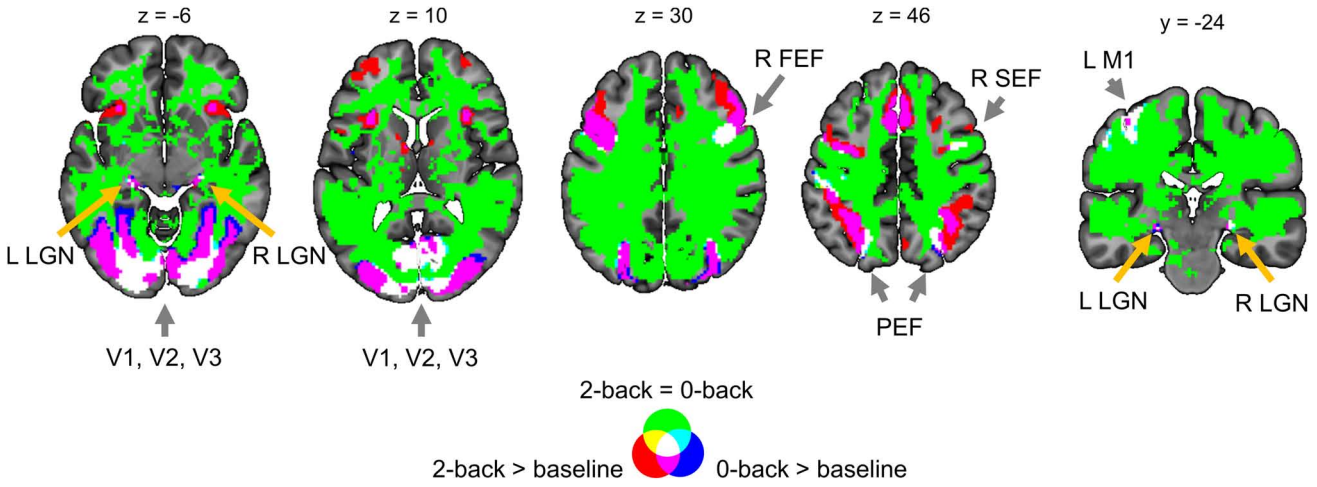


FIGURE 19 | Example of possible application of BPI based on the working memory task. L/R, left/right; V1, V2, V3, primary, secondary, and third visual cortex; FEF, frontal eye field; SEF, superior eye field; PEF, parietal eye field; LGN, lateral geniculate nucleus; M1, primary motor cortex.

reach the plateau and still label the brain area as 'low confidence.' However, even in this case, we can make a definite conclusion: the task design is not sensitive to the effect and should be revised. Empirical Bayes with the 'global shrinkage' prior allows us to monitor the evidence for the alternative or null hypotheses after each participant without special adjustment for multiplicity (Edwards et al., 1963; Berger and Berry, 1988; Wagenmakers, 2007; Rouder, 2014; Kruschke and Liddell, 2017b; Schönbrodt et al., 2017). The optional stopping of the experiment not only allows more freedom in terms of the experimental design, but also saves limited resources and is even more ethically justified in certain cases[6] (Edwards et al., 1963; Wagenmakers, 2007). To strike a balance between analytical flexibility and subjectivity of analysis, one may pre-register hypotheses, models, priors and desired level of evidence to reach without being limited by predefined sample size.

In contrast, frequentist inference depends on the researcher's intention to stop data collection and thus requires a definition of the stopping rule based on *a priori* power analysis. The sequential analysis and optional stopping in frequentist inference inflate the number of false positives and require special multiplicity adjustments. Moreover, even if the *a priori* defined sample size is reached, the researcher can still obtain a non-significant result. In this case, the researcher can follow two controversial paths within the classical NHST framework. Firstly, the sample size could be further increased to force an indecisive result to a decisive conclusion. The problem is that this conclusion would always be against the null hypothesis. Thus, an unbounded increase in the sample size introduces a discrepancy between classical NHST and Bayesian inference, also known as the Jeffreys-Lindley paradox. Secondly, one may argue that high *a priori* power and non-significant results provide evidence for the null hypothesis (see, for example, Cohen, 1990). However, even high *a priori* power and non-significant results do not provide direct evidence for the null hypothesis. In fact, a high-powered non-significant result may arise when the obtained data provide no evidence for the null over the alternative hypothesis, according to Bayesian inference (Dienes and Mclatchie, 2017). This does not mean that power analysis is irrelevant from a Bayesian perspective. Although power analysis is not necessary for Bayesian inference, it can still be used within the Bayesian framework for study planning (Kruschke and Liddell, 2017b). At the same time, power analysis is a critical part of frequentist inference, as it depends on researcher intentions, such as the stopping intention.

The main difficulty with the application of BPI is the need to define the ES threshold. However, the problem of choosing a practically meaningful effect size is not unique to fMRI studies, as it arises in every mature field of science. It should not discourage us from using BPI, as the point-null hypothesis is never true in the soft sciences. From our perspective, there are several ways to address this problem. Firstly, the ES threshold can be chosen based on previously reported effect sizes in studies with a similar design or perform a pilot study to estimate the expected effect size.

Based on the fMRI literature, the largest BOLD responses are evoked by sensory stimulation and vary within 1–5% of the overall mean whole-brain activity. In contrast, BOLD responses induced by cognitive tasks vary within 0.1–0.5% (Friston et al., 2002b; Poldrack et al., 2011; Chen et al., 2017). The results obtained in this study support this notion. Primary sensory effects were >1%, and motor effects were >0.3%. Cognitive effects can be classified into three categories.

(1) 'Strong' effects of 0.2−0.3% (for example, emotion processing in the amygdala, language processing in Broca's area),

[6]This is especially true for PET studies. The BPI method described in this work can also be applied to PET data to reduce the sample size and thus exposure to radioactivity (Svensson et al., 2020).

(2) 'Moderate' effects of 0.1−0.2% (for example, working memory load in DLPFC, social cognition in IPL, response inhibition in IFG/FO),
(3) 'Weak' effects of 0.05–0.1% in contrasts without robust activations (for example, reward processing in the nucleus accumbens, relational processing in DLPFC).

However, choosing the ES threshold based on previous studies can be challenging because fMRI designs become increasingly complex over time, and it can be difficult to find previous experiments reporting unbiased effect size with a similar design. In this case, one can use the ES threshold equal to *one prior SD* of the effect (Friston and Penny, 2003), which can be thought as a neuronal 'background noise level' or a level of activity that is generic to the whole brain (Eickhoff et al., 2008). As empirical and simulation analysis results show, BPI with this ES threshold generally works well for both 'activated/deactivated' and 'not activated' voxel detection. However, it may not be suitable in cases with the weak effects and high noise. In addition, researchers who rely more on the frequentist inference may use the $\gamma(Dice_{max})$ threshold to replicate the results obtained previously with classical NHST and additionally search for 'not activated' and 'low confidence' voxels. Finally, the degree to which the posterior probability is contained within the ROPEs of different widths could be specified or the ROPE maps in which the thresholding sequence is inverted could be calculated. The ROPE maps can be shared in public repositories, such as Neurovault, along with PPMs, and subsequently thresholded by any reasonable ES threshold.

The ability to provide evidence for the null hypothesis may be especially beneficial for clinical neuroimaging. Possible issues that can be resolved using this approach are:

(1) Let the brain activity in certain ROIs due to a neurodegenerative process decrease by more than γ per year on average without any treatment. To prove that a new treatment *effectively protects against neurodegenerative processes*, we can provide evidence that, within 1 year of treatment, brain activity was reduced by less than X%.
(2) Assume that an effective treatment should change the brain activity in certain ROIs by at least X%. Then, we can prove that a new treatment is *practically ineffective* if the activity has changed by less than X%.
(3) Consider two groups of subjects taking a new treatment and a placebo, respectively. Using BPI, we can provide evidence that the result of the new treatment is *does not differ from that of the placebo*.
(4) Consider two groups of subjects taking an old effective treatment and a new treatment. Using BPI, we can provide evidence that the new treatment is *no worse than the old effective treatment*.
(5) Consider a new treatment for a disease that *is not related to brain function*. Using BPI, we can provide evidence that the new treatment *does not have side effects* on brain activity.

CONCLUSION

Herein, a discussion of the use of the Bayesian and frequentist approaches to assess the 'null effects' in fMRI studies was presented. We demonstrated that group-level Bayesian inference may be more intuitive and convenient in practice than frequentist inference. Crucially, Bayesian inference can detect 'activated/deactivated,' 'not activated,' and 'low confidence' voxels using a single decision rule. Moreover, it allows for interim analysis and optional stopping when the obtained sample size is sufficient to make a confident inference. We considered the problem of defining a threshold for the effect size and provided a reference set of typical effect sizes in different fMRI designs. Bayesian inference and assessment of the 'null effects' may be especially beneficial for basic and applied clinical neuroimaging. The developed SPM12-based toolbox with a simple GUI is expected to be useful for the assessment of 'null effects' using BPI.

LIMITATIONS AND FUTURE WORK

Firstly, we did not consider BMI, which is currently mainly used for the analysis of effective connectivity. A promising area of future research would be to compare the advantages of BMI and BPI when analyzing local brain activity. Secondly, the 'global shrinkage' prior must be compared with other possible priors, in particular with priors that take into account the spatial dependency between voxels. Thirdly, we used Bayesian statistics only at the group level. Future studies could consider the advantages of using the Bayesian approach at both the subject and group levels.

DATA AVAILABILITY STATEMENT

The datasets analyzed for this study can be found in the Human Connectome Project (https://www.humanconnectome.org/study/hcp-young-adult/document/1200-subjects-data-release) and the UCLA Consortium for Neuropsychiatric Phenomics study (https://openneuro.org/datasets/ds000030/versions/1.0.0). Bayesian parameter inference was performed using the SPM12-based toolbox available at https://github.com/Masharipov/Bayesian_inference.

AUTHOR CONTRIBUTIONS

RM, AK, and MK contributed to conceptualization of the research. MK supervised the project. RM, IK, and YN contributed to statistical analysis and programming. RM and IK performed simulations. MD, DC, and MK acquired funding. All authors contributed to the text of this article and approved the submitted version.

FUNDING

RM, AK, and MK were supported by the Russian Science Foundation Grant #19-18-00454. IK, YN, MD, and DC were supported by the state assignment of the Ministry of Education and Science of Russian Federation (theme number AAAA-A19-119101890066-2). Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University. Another part of the data were provided by UCLA dataset which was obtained from the OpenfMRI database (its accession number is ds000030) and data collection was funded by the Consortium for Neuropsychiatric Phenomics (NIH Roadmap for Medical Research grants UL1-DE019580, RL1MH083268, RL1MH083269, RL1DA024853, RL1MH083270, RL1LM009833, PL1MH083271, and PL1NS062410).

ACKNOWLEDGMENTS

We thank Andrey Ogai for the valuable help with a code for visualization of statistical maps.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fninf.2021.738342/full#supplementary-material

REFERENCES

Acar, F., Seurinck, R., Eickhoff, S. B., and Moerkerke, B. (2018). Assessing robustness against potential publication bias in activation likelihood estimation (ALE) meta-analyses for fMRI. *PLoS One* 13:e0208177. doi: 10.1371/journal.pone.0208177

Aczel, B., Palfi, B., Szollosi, A., Kovacs, M., Szaszi, B., Szecsi, P., et al. (2018). Quantifying support for the null hypothesis in psychology: an empirical investigation. *Adv. Methods Pract. Psychol. Sci.* 1, 357–366. doi: 10.1177/2515245918773742

Alberton, B. A., Nichols, T. E., Gamba, H. R., and Winkler, A. M. (2020). Multiple testing correction over contrasts for brain imaging. *Neuroimage* 216:116760. doi: 10.1016/j.neuroimage.2020.116760

Altman, D. G., and Bland, J. M. (1995). Statistics notes: absence of evidence is not evidence of absence. *BMJ* 311:485. doi: 10.1136/bmj.311.7003.485

Amrhein, V., Korner-Nievergelt, F., and Roth, T. (2017). The earth is flat ($p > 0.05$): significance thresholds and the crisis of unreplicable research. *PeerJ* 5:e3544. doi: 10.7717/peerj.3544

Baguley, T. (2009). Standardized or simple effect size: what should be reported? *Br. J. Psychol.* 100, 603–617. doi: 10.1348/000712608x377117

Barch, D. M., Burgess, G. C., Harms, M. P., Petersen, S. E., Schlaggar, B. L., Corbetta, M., et al. (2013). Function in the human connectome: task-fMRI and individual differences in behavior. *Neuroimage* 80, 169–189. doi: 10.1016/j.neuroimage.2013.05.033

Belia, S., Fidler, F., Williams, J., and Cumming, G. (2005). Researchers misunderstand confidence intervals and standard error bars. *Psychol. Methods* 10, 389–396. doi: 10.1037/1082-989x.10.4.389

Berger, J. O. (2003). Could Fisher, Jeffreys and Neyman have agreed on testing? *Stat. Sci.* 18, 1–32. doi: 10.1214/ss/1056397485

Berger, J. O., and Berry, D. A. (1988). Statistical analysis and the illusion of objectivity. *Am. Sci.* 76, 159–165.

Berger, J. O., and Sellke, T. (1987). Testing a point null hypothesis: the irreconcilability of p values and evidence: rejoinder. *J. Am. Stat. Assoc.* 82:135. doi: 10.2307/2289139

Berry, D. (1988). "Multiple comparisons, multiple tests, and data dredging: a Bayesian perspective," in *Bayesian Statistics*, eds J. Bernardo, M. DeGroot, D. Lindley, and A. Smith (Oxford: Oxford University Press), 79–94.

Berry, D. A., and Hochberg, Y. (1999). Bayesian perspectives on multiple comparisons. *J. Stat. Plan. Inference* 82, 215–227. doi: 10.1016/s0378-3758(99)00044-0

Campbell, H., and Gustafson, P. (2018). Conditional equivalence testing: an alternative remedy for publication bias. *PLoS One* 13:e0195145. doi: 10.1371/journal.pone.0195145

Chen, G., Cox, R. W., Glen, D. R., Rajendra, J. K., Reynolds, R. C., and Taylor, P. A. (2018). A tail of two sides: artificially doubled false positive rates in neuroimaging due to the sidedness choice with t -tests. *Hum. Brain Mapp.* 40, 1037–1043. doi: 10.1002/hbm.24399

Chen, G., Taylor, P. A., and Cox, R. W. (2017). Is the statistic value all we should care about in neuroimaging? *Neuroimage* 147, 952–959. doi: 10.1016/j.neuroimage.2016.09.066

Chen, G., Taylor, P. A., Cox, R. W., and Pessoa, L. (2020). Fighting or embracing multiplicity in neuroimaging? Neighborhood leverage versus global calibration. *Neuroimage* 206:116320. doi: 10.1016/j.neuroimage.2019.116320

Chen, G., Xiao, Y., Taylor, P. A., Rajendra, J. K., Riggins, T., Geng, F., et al. (2019). Handling multiplicity in neuroimaging through Bayesian lenses with multilevel modeling. *Neuroinformatics* 17, 515–545. doi: 10.1007/s12021-018-9409-6

Cohen, J. (1965). "Some statistical issues in psychological research," in *Handbook of Clinical Psychology*, ed. B. B. Wolman (New York, NY: McGraw-Hill), 95–121.

Cohen, J. (1990). Things I have learned (so far). *Am. Psychol.* 45, 1304–1312. doi: 10.1037/0003-066x.45.12.1304

Cohen, J. (1994). The earth is round ($p < .05$). *Am. Psychol.* 49, 997–1003. doi: 10.1037/0003-066x.49.12.997

Cornfield, J. (1966). Sequential trials, sequential analysis and the likelihood principle. *Am. Stat.* 20, 18–23. doi: 10.1080/00031305.1966.10479786

Cortina, J. M., and Dunlap, W. P. (1997). On the logic and purpose of significance testing. *Psychol. Methods* 2, 161–172. doi: 10.1037/1082-989x.2.2.161

Cramer, A. O. J., van Ravenzwaaij, D., Matzke, D., Steingroever, H., Wetzels, R., Grasman, R. P. P. P., et al. (2015). Hidden multiplicity in exploratory multiway ANOVA: prevalence and remedies. *Psychon. Bull. Rev.* 23, 640–647. doi: 10.3758/s13423-015-0913-5

Cremers, H. R., Wager, T. D., and Yarkoni, T. (2017). The relation between statistical power and inference in fMRI. *PLoS One* 12:e0184923. doi: 10.1371/journal.pone.0184923

Cumming, G. (2013). The new statistics: why and how. *Psychol. Sci.* 25, 7–29. doi: 10.1177/0956797613504966

Dandolo, L. C., and Schwabe, L. (2019). Time-dependent motor memory representations in prefrontal cortex. *Neuroimage* 197, 143–155. doi: 10.1016/j.neuroimage.2019.04.051

David, S. P., Naudet, F., Laude, J., Radua, J., Fusar-Poli, P., Chu, I., et al. (2018). Potential reporting bias in neuroimaging studies of sex differences. *Sci. Rep.* 8:6082. doi: 10.1038/s41598-018-23976-1

de Winter, J. C., and Dodou, D. (2015). A surge ofp-values between 0.041 and 0.049 in recent decades (but negative results are increasing rapidly too). *PeerJ* 3:e733. doi: 10.7717/peerj.733

Dienes, Z. (2014). Using Bayes to get the most out of non-significant results. *Front. Psychol.* 5:781. doi: 10.3389/fpsyg.2014.00781

Dienes, Z., and Mclatchie, N. (2017). Four reasons to prefer Bayesian analyses over significance testing. *Psychon. Bull. Rev.* 25, 207–218. doi: 10.3758/s13423-017-1266-z

Edwards, W., Lindman, H., and Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychol. Rev.* 70, 193–242. doi: 10.1037/h0044139

Eickhoff, S. B., Grefkes, C., Fink, G. R., and Zilles, K. (2008). Functional lateralization of face, hand, and trunk representation in anatomically defined

human somatosensory areas. *Cereb. Cortex* 18, 2820–2830. doi: 10.1093/cercor/bhn039

Eklund, A., Nichols, T. E., and Knutsson, H. (2016). Cluster failure: why fMRI inferences for spatial extent have inflated false-positive rates. *Proc. Natl. Acad. Sci. U.S.A.* 113, 7900–7905. doi: 10.1073/pnas.1602413113

Falk, R., and Greenbaum, C. W. (1995). Significance tests die hard. *Theory Psychol.* 5, 75–98. doi: 10.1177/0959354395051004

Feng, C., Forthman, K. L., Kuplicki, R., Yeh, H. W., Stewart, J. L., and Paulus, M. P. (2019). Neighborhood affluence is not associated with positive and negative valence processing in adults with mood and anxiety disorders: a Bayesian inference approach. *Neuroimage Clin.* 22:101738. doi: 10.1016/j.nicl.2019.101738

Fidler, F., Burgman, M. A., Cumming, G., Buttrose, R., and Thomason, N. (2006). Impact of criticism of null-hypothesis significance testing on statistical reporting practices in conservation biology. *Conserv. Biol.* 20, 1539–1544. doi: 10.1111/j.1523-1739.2006.00525.x

Finch, S., Cumming, G., and Thomason, N. (2001). Colloquium on effect sizes: the roles of editors, textbook authors, and the publication manual. *Educ. Psychol. Meas.* 61, 181–210. doi: 10.1177/0013164401612001

Friston, K. (2012). Ten ironic rules for non-statistical reviewers. *Neuroimage* 61, 1300–1310. doi: 10.1016/j.neuroimage.2012.04.018

Friston, K. (2013). Sample size and the fallacies of classical inference. *Neuroimage* 81, 503–504. doi: 10.1016/j.neuroimage.2013.02.057

Friston, K., Penny, W., Phillips, C., Kiebel, S., Hinton, G., and Ashburner, J. (2002a). Classical and Bayesian inference in neuroimaging: theory. *Neuroimage* 16, 465–483. doi: 10.1006/nimg.2002.1090

Friston, K., Glaser, D., Henson, R., Kiebel, S., Phillips, C., and Ashburner, J. (2002b). Classical and Bayesian inference in neuroimaging: applications. *Neuroimage* 16, 484–512. doi: 10.1006/nimg.2002.1091

Friston, K., Mattout, J., Trujillo-Barreto, N., Ashburner, J., and Penny, W. (2007). Variational free energy and the Laplace approximation. *Neuroimage* 34, 220–234. doi: 10.1016/j.neuroimage.2006.08.035

Friston, K., and Penny, W. (2003). Posterior probability maps and SPMs. *Neuroimage* 19, 1240–1249. doi: 10.1016/s1053-8119(03)00144-7

Friston, K., and Penny, W. (2011). Post hoc Bayesian model selection. *Neuroimage* 56, 2089–2099. doi: 10.1016/j.neuroimage.2011.03.062

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

Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S. J., and Turner, R. (1996). Movement-related effects in fMRI time-series. *Magn. Reson. Med.* 35, 346–355. doi: 10.1002/mrm.1910350312

Gelman, A., Hill, J., and Yajima, M. (2012). Why we (usually) don't have to worry about multiple comparisons. *J. Res. Educ. Eff.* 5, 189–211. doi: 10.1080/19345747.2011.618213

Genovese, C. R. (2000). A Bayesian time-course model for functional magnetic resonance imaging data: rejoinder. *J. Am. Stat. Assoc.* 95:716. doi: 10.2307/2669451

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

Gigerenzer, G. (1993). "The superego, the ego, and the id in statistical reasoning," in *A Handbook for Data Analysis in the Behavioral Sciences: Methodological Issues*, eds G. Keren and C. Lewis (Mahwah, NJ: Lawrence Erlbaum Associates, Inc.), 311–339.

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., et al. (2013). The minimal preprocessing pipelines for the human connectome project. *Neuroimage* 80, 105–124. doi: 10.1016/j.neuroimage.2013.04.127

Gonzalez-Castillo, J., Saad, Z. S., Handwerker, D. A., Inati, S. J., Brenowitz, N., and Bandettini, P. A. (2012). Whole-brain, time-locked activation with simple tasks revealed using massive averaging and model-free analysis. *Proc. Natl. Acad. Sci. US.A.* 109, 5487–5492. doi: 10.1073/pnas.1121049109

Goodman, S. (2008). A dirty dozen: twelve P-value misconceptions. *Semin. Hematol.* 45, 135–140. doi: 10.1053/j.seminhematol.2008.04.003

Goodman, S. N. (1993). p values, hypothesis tests, and likelihood: implications for epidemiology of a neglected historical debate. *Am. J. Epidemiol.* 137, 485–496. doi: 10.1093/oxfordjournals.aje.a116700

Gopalan, R., and Berry, D. A. (1998). Bayesian multiple comparisons using dirichlet process priors. *J. Am. Stat. Assoc.* 93, 1130–1139. doi: 10.1080/01621459.1998.10473774

Gorgolewski, K. J., Durnez, J., and Poldrack, R. A. (2017). Preprocessed consortium for neuropsychiatric phenomics dataset. *F1000Res.* 6:1262. doi: 10.12688/f1000research.11964.2

Greenland, S. (2019). Valid P-values behave exactly as they should: some misleading criticisms of P-values and their resolution with S-values. *Am. Stat.* 73(Suppl. 1), 106–114. doi: 10.1080/00031305.2018.1529625

Greenland, S., Senn, S. J., Rothman, K. J., Carlin, J. B., Poole, C., Goodman, S. N., et al. (2016). Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations. *Eur. J. Epidemiol.* 31, 337–350. doi: 10.1007/s10654-016-0149-3

Greenwald, A. G. (1975). Consequences of prejudice against the null hypothesis. *Psychol. Bull.* 82, 1–20. doi: 10.1037/h0076157

Gusnard, D. A., and Raichle, M. E. (2001). Searching for a baseline: functional imaging and the resting human brain. *Nat. Rev. Neurosci.* 2, 685–694. doi: 10.1038/35094500

Hagen, R. L. (1997). In praise of the null hypothesis statistical test. *Am. Psychol.* 52, 15–24. doi: 10.1037/0003-066x.52.1.15

Hodges, J. L., and Lehmann, E. L. (1954). Testing the approximate validity of statistical hypotheses. *J. R. Stat. Soc. Ser. B Methodol.* 16, 261–268. doi: 10.1111/j.2517-6161.1954.tb00169.x

Hoekstra, R., Finch, S., Kiers, H. A. L., and Johnson, A. (2006). Probability as certainty: dichotomous thinking and the misuse of p values. *Psychon. Bull. Rev.* 13, 1033–1037. doi: 10.3758/bf03213921

Hoekstra, R., Morey, R. D., Rouder, J. N., and Wagenmakers, E. J. (2014). Robust misinterpretation of confidence intervals. *Psychon. Bull. Rev.* 21, 1157–1164. doi: 10.3758/s13423-013-0572-3

Hubbard, R., and Bayarri, M. J. (2003). Confusion over measures of evidence (p's) versus errors (α's) in classical statistical testing. *Am. Stat.* 57, 171–178. doi: 10.1198/0003130031856

Hubbard, R., and Lindsay, R. M. (2008). Why p values are not a useful measure of evidence in statistical significance testing. *Theory Psychol.* 18, 69–88. doi: 10.1177/0959354307086923

Ioannidis, J. P., Munafò, M. R., Fusar-Poli, P., Nosek, B. A., and David, S. P. (2014). Publication and other reporting biases in cognitive sciences: detection, prevalence, and prevention. *Trends Cogn. Sci.* 18, 235–241. doi: 10.1016/j.tics.2014.02.010

Ioannidis, J. P. A. (2019). What have we (not) learnt from millions of scientific papers with p values? *Am. Stat.* 73(Suppl. 1), 20–25. doi: 10.1080/00031305.2018.1447512

Jeffreys, H. (1939/1948). *Theory of Probability*, 2nd Edn. Oxford: The Clarendon Press.

Jennings, R. G., and Van Horn, J. D. (2012). Publication bias in neuroimaging research: implications for meta-analyses. *Neuroinformatics* 10, 67–80. doi: 10.1007/s12021-011-9125-y

Johansson, T. (2011). Hail the impossible: p-values, evidence, and likelihood. *Scand. J. Psychol.* 52, 113–125. doi: 10.1111/j.1467-9450.2010.00852.x

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994). *Continuous Univariate Distributions*, Vol. 6. New York, NY: John Wiley and Sons, 1–119.

Joyce, K. E., and Hayasaka, S. (2012). Development of PowerMap: a software package for statistical power calculation in neuroimaging studies. *Neuroinformatics* 10, 351–365. doi: 10.1007/s12021-012-9152-3

Kass, R. E., and Raftery, A. E. (1995). Bayes factors. *J. Am. Stat. Assoc.* 90, 773–795. doi: 10.1080/01621459.1995.10476572

Kirk, R. E. (1996). Practical significance: a concept whose time has come. *Educ. Psychol. Meas.* 56, 746–759. doi: 10.1177/0013164496056005002

Knief, U., and Forstmeier, W. (2021). Violating the normality assumption may be the lesser of two evils. *Behav. Res. Methods* 1–15. doi: 10.3758/s13428-021-01587-5

Kruschke, J. K. (2010). What to believe: Bayesian methods for data analysis. *Trends Cogn. Sci.* 14, 293–300. doi: 10.1016/j.tics.2010.05.001

Kruschke, J. K. (2011). Bayesian assessment of null values via parameter estimation and model comparison. *Perspect. Psychol. Sci.* 6, 299–312. doi: 10.1177/1745691611406925

Kruschke, J. K., and Liddell, T. M. (2017b). The Bayesian new statistics: hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychon. Bull. Rev.* 25, 178–206. doi: 10.3758/s13423-016-1221-4

Kruschke, J. K., and Liddell, T. M. (2017a). Bayesian data analysis for newcomers. *Psychon. Bull. Rev.* 25, 155–177. doi: 10.3758/s13423-017-1272-1

Lakens, D. (2017). Equivalence tests. *Soc. Psychol. Pers. Sci.* 8, 355–362. doi: 10.1177/1948550617697177

Lakens, D., McLatchie, N., Isager, P. M., Scheel, A. M., and Dienes, Z. (2018). Improving inferences about null effects with Bayes factors and equivalence tests. *J. Gerontol. Ser. B* 75, 45–57. doi: 10.1093/geronb/gby065

Liao, J. G., Midya, V., and Berg, A. (2019). Connecting Bayes factor and the region of practical equivalence (ROPE) procedure for testing interval null hypothesis. *arXiv* [Preprint] arXiv:1903.03153,

Lindley, D. (1965). *Introduction to Probability and Statistics from a Bayesian Viewpoint*, 1st Edn. Cambridge: Cambridge University Press.

Lindley, D. V. (1957). A statistical paradox. *Biometrika* 44:187. doi: 10.2307/2333251

Lindley, D. V. (1975). The future of statistics: a Bayesian 21st century. *Adv. Appl. Probab.* 7:106. doi: 10.2307/1426315

Lindley, D. V. (1990). The 1988 wald memorial lectures: the present position in Bayesian statistics. *Stat. Sci.* 5, 44–65. doi: 10.1214/ss/1177012253

Magerkurth, J., Mancini, L., Penny, W., Flandin, G., Ashburner, J., Micallef, C., et al. (2015). Objective Bayesian fMRI analysis–a pilot study in different clinical environments. *Front. Neurosci.* 9:168. doi: 10.3389/fnins.2015.00168

Meehl, P. E. (1967). Theory-testing in psychology and physics: a methodological paradox. *Philos. Sci.* 34, 103–115. doi: 10.1086/288135

Meehl, P. E. (1978). Theoretical risks and tabular asterisks: Sir Karl, Sir Ronald, and the slow progress of soft psychology. *J. Consult. Clin. Psychol.* 46, 806–834. doi: 10.1037/0022-006X.46.4.806

Meyners, M. (2012). Equivalence tests – a review. *Food Qual. Prefer.* 26, 231–245. doi: 10.1016/j.foodqual.2012.05.003

Morey, R. D., Hoekstra, R., Rouder, J. N., and Wagenmakers, E. J. (2015). Continued misinterpretation of confidence intervals: response to Miller and Ulrich. *Psychon. Bull. Rev.* 23, 131–140. doi: 10.3758/s13423-015-0955-8

Morey, R. D., and Rouder, J. N. (2011). Bayes factor approaches for testing interval null hypotheses. *Psychol. Methods* 16, 406–419. doi: 10.1037/a0024377

Muller, P., Parmigiani, G., and Rice, K. (2006). "FDR and Bayesian multiple comparisons rules," in *Proceedings of the 8th Valencia International Meeting Bayesian Statistics 8*, eds J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, et al. (Oxford: Oxford University Press), 366–368.

Mumford, J. A. (2012). A power calculation guide for fMRI studies. *Soc. Cogn. Affect. Neurosci.* 7, 738–742. doi: 10.1093/scan/nss059

Mumford, J. A., and Nichols, T. E. (2008). Power calculation for group fMRI studies accounting for arbitrary design and temporal autocorrelation. *Neuroimage* 39, 261–268. doi: 10.1016/j.neuroimage.2007.07.061

Murphy, K. R., and Myors, B. (1999). Testing the hypothesis that treatments have negligible effects: minimum-effect tests in the general linear model. *J. Appl. Psychol.* 84, 234–248. doi: 10.1037/0021-9010.84.2.234

Murphy, K. R., and Myors, B. (2004). *Statistical Power Analysis: A Simple and General Model for Traditional and Modern Hypothesis Tests*, 2nd Edn. Mahwah, NJ: Lawrence Erlbaum Associates.

Nichols, T., and Hayasaka, S. (2003). Controlling the familywise error rate in functional neuroimaging: a comparative review. *Stat. Methods Med. Res.* 12, 419–446. doi: 10.1191/0962280203sm341ra

Nichols, T. E. (2012). Multiple testing corrections, nonparametric methods, and random field theory. *Neuroimage* 62, 811–815. doi: 10.1016/j.neuroimage.2012.04.014

Nickerson, R. S. (2000). Null hypothesis significance testing: a review of an old and continuing controversy. *Psychol. Methods* 5, 241–301. doi: 10.1037/1082-989x.5.2.241

Penny, W., Flandin, G., and Trujillo-Barreto, N. (2007). Bayesian comparison of spatially regularised general linear models. *Hum. Brain Mapp.* 28, 275–293. doi: 10.1002/hbm.20327

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

Penny, W. D., and Ridgway, G. R. (2013). Efficient posterior probability mapping using savage-dickey ratios. *PLoS One* 8:e59655. doi: 10.1371/journal.pone.0059655

Penny, W. D., Trujillo-Barreto, N. J., and Friston, K. J. (2005). Bayesian fMRI time series analysis with spatial priors. *Neuroimage* 24, 350–362. doi: 10.1016/j.neuroimage.2004.08.034

Perezgonzalez, J. D. (2015). Fisher, Neyman-Pearson or NHST? A tutorial for teaching data testing. *Front. Psychol.* 6:223. doi: 10.3389/fpsyg.2015.00223

Pernet, C. R. (2014). Misconceptions in the use of the general linear model applied to functional MRI: a tutorial for junior neuro-imagers. *Front. Neurosci.* 8:1. doi: 10.3389/fnins.2014.00001

Poldrack, R., Congdon, E., Triplett, W., Gorgolewski, K., Karlsgodt, K., Mumford, J., et al. (2016). A phenome-wide examination of neural and cognitive function. *Sci. Data* 3:160110. doi: 10.1038/sdata.2016.110

Poldrack, R. A., Baker, C. I., Durnez, J., Gorgolewski, K. J., Matthews, P. M., Munafò, M. R., et al. (2017). Scanning the horizon: towards transparent and reproducible neuroimaging research. *Nat. Rev. Neurosci.* 18, 115–126. doi: 10.1038/nrn.2016.167

Poldrack, R. A., Mumford, J. A., and Nichols, T. E. (2011). *Handbook of Functional MRI Data Analysis*. Cambridge: Cambridge University Press.

Poline, J. B., and Brett, M. (2012). The general linear model and fMRI: does love last forever? *Neuroimage* 62, 871–880. doi: 10.1016/j.neuroimage.2012.01.133

Pollard, P., and Richardson, J. T. (1987). On the probability of making type I errors. *Psychol. Bull.* 102, 159–163. doi: 10.1037/0033-2909.102.1.159

Raichle, M. E., and Gusnard, D. A. (2002). Appraising the brain's energy budget. *Proc. Natl. Acad. Sci. U.S.A.* 99, 10237–10239. doi: 10.1073/pnas.172399499

Reimold, M., Slifstein, M., Heinz, A., Mueller-Schauenburg, W., and Bares, R. (2005). Effect of spatial smoothing on t-Maps: arguments for going back from t-Maps to masked contrast images. *J. Cereb. Blood Flow Metab.* 26, 751–759. doi: 10.1038/sj.jcbfm.9600231

Rogers, J. L., Howard, K. I., and Vessey, J. T. (1993). Using significance tests to evaluate equivalence between two experimental groups. *Psychol. Bull.* 113, 553–565. doi: 10.1037/0033-2909.113.3.553

Rosa, M., Friston, K., and Penny, W. (2012). Post-hoc selection of dynamic causal models. *J. Neurosci. Methods* 208, 66–78. doi: 10.1016/j.jneumeth.2012.04.013

Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychol. Bull.* 86, 638–641. doi: 10.1037/0033-2909.86.3.638

Rouder, J. N. (2014). Optional stopping: no problem for Bayesians. *Psychon. Bull. Rev.* 21, 301–308. doi: 10.3758/s13423-014-0595-4

Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., and Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychon. Bull. Rev.* 16, 225–237. doi: 10.3758/pbr.16.2.225

Royall, R. M. (1986). The effect of sample size on the meaning of significance tests. *Am. Stat.* 40:313. doi: 10.2307/2684616

Royall, R. M. (1997). *Statistical Evidence: A Likelihood Paradigm*. Boca Raton, FL: CRC Press.

Samartsidis, P., Montagna, S., Laird, A. R., Fox, P. T., Johnson, T. D., and Nichols, T. E. (2020). Estimating the prevalence of missing experiments in a neuroimaging meta-analysis. *Res. Synth. Methods* 11, 866–883. doi: 10.1002/jrsm.1448

Schatz, P., Jay, K., McComb, J., and McLaughlin, J. (2005). Misuse of statistical tests in publications. *Arch. Clin. Neuropsychol.* 20, 1053–1059. doi: 10.1016/j.acn.2005.06.006

Schneider, J. W. (2014). Null hypothesis significance tests. A mix-up of two different theories: the basis for widespread confusion and numerous misinterpretations. *Scientometrics* 102, 411–432. doi: 10.1007/s11192-014-1251-5

Schneider, J. W. (2018). NHST is still logically flawed. *Scientometrics* 115, 627–635. doi: 10.1007/s11192-018-2655-4

Schönbrodt, F. D., Wagenmakers, E.-J., Zehetleitner, M., and Perugini, M. (2017). Sequential hypothesis testing with Bayes factors: Efficiently testing mean differences. *Psychol. Methods* 22, 322–339. doi: 10.1037/met0000061

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *J. Pharmacokinet. Biopharm.* 15, 657–680. doi: 10.1007/bf01068419

Schwartzman, A., Dougherty, R., Lee, J., Ghahremani, D., and Taylor, J. (2009). Empirical null and false discovery rate analysis in neuroimaging. *Neuroimage* 44, 71–82. doi: 10.1016/j.neuroimage.2008.04.182

Serlin, R. C., and Lapsley, D. K. (1985). Rationality in psychological research: the good-enough principle. *Am. Psychol.* 40, 73–83. doi: 10.1037/0003-066x.40.1.73

Serlin, R. C., and Lapsley, D. K. (1993). "Rational appraisal of psychological research and the good-enough principle," in *A Handbook for Data Analysis in the Behavioral Sciences: Methodological Issues*, eds G. Keren and C. Lewis (Mahwah, NJ: Lawrence Erlbaum Associates, Inc.), 199–228.

Shapiro, S. S., and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika* 52, 591–611. doi: 10.1093/biomet/52.3-4.591

Sjölander, A., and Vansteelandt, S. (2019). Frequentist versus Bayesian approaches to multiple testing. *Eur. J. Epidemiol.* 34, 809–821. doi: 10.1007/s10654-019-00517-2

Smith, S. M., and Nichols, T. E. (2018). Statistical challenges in "big data" human neuroimaging. *Neuron* 97, 263–268. doi: 10.1016/j.neuron.2017.12.018

Sober, E. (2008). *Evidence and Evolution: The Logic Behind the Science*. Cambridge: Cambridge University Press.

Storey, J. D. (2003). The positive false discovery rate: a Bayesian interpretation and the q-value. *Ann. Stat.* 31, 2013–2035. doi: 10.1214/aos/1074290335

Streiner, D. L. (2015). Best (but oft-forgotten) practices: the multiple problems of multiplicity—whether and how to correct for many statistical tests. *Am. J. Clin. Nutr.* 102, 721–728. doi: 10.3945/ajcn.115.113548

Svensson, J., Schain, M., Knudsen, G. M., Ogden, T., and Plavén-Sigray, P. (2020). Early stopping in clinical PET studies: how to reduce expense and exposure. *MedRxiv* [Preprint] doi: 10.1101/2020.09.13.20192856

Szucs, D., and Ioannidis, J. P. (2020). Sample size evolution in neuroimaging research: an evaluation of highly-cited studies (1990–2012) and of latest practices (2017–2018) in high-impact journals. *Neuroimage* 221:117164. doi: 10.1016/j.neuroimage.2020.117164

Szucs, D., and Ioannidis, J. P. A. (2017). When null hypothesis significance testing is unsuitable for research: a reassessment. *Front. Hum. Neurosci.* 11:390. doi: 10.3389/fnhum.2017.00390

Turkheimer, F. E., Aston, J. A. D., and Cunningham, V. J. (2004). On the logic of hypothesis testing in functional imaging. *Eur. J. Nuclear Med. Mol. Imaging* 31, 725–732. doi: 10.1007/s00259-003-1387-7

UIudag, K., Müller-Bierl, B., and Ugurbil, K. (2009). An integrative model for neuronal activity-induced signal changes for gradient and spin echo functional imaging. *Neuroimage* 47:S56. doi: 10.1016/s1053-8119(09)70204-6

Wagenmakers, E. J. (2007). A practical solution to the pervasive problems of p values. *Psychon. Bull. Rev.* 14, 779–804. doi: 10.3758/bf03194105

Wagenmakers, E. J., Lee, M., Lodewyckx, T., and Iverson, G. J. (2008). "Bayesian versus Frequentist inference," in *Bayesian Evaluation of Informative Hypotheses. Statistics for Social and Behavioral Sciences*, eds H. Hoijtink, I. Klugkist, and P. A. Boelen (New York, NY: Springer), 181–207.

Wagenmakers, E. J., Lodewyckx, T., Kuriyal, H., and Grasman, R. (2010). Bayesian hypothesis testing for psychologists: a tutorial on the Savage–Dickey method. *Cogn. Psychol.* 60, 158–189. doi: 10.1016/j.cogpsych.2009.12.001

Wagenmakers, E. J., Verhagen, J., Ly, A., Matzke, D., Steingroever, H., Rouder, J. N., et al. (2017). "The need for Bayesian Hypothesis testing in psychological science," in *Psychological Science Under Scrutiny: Recent Challenges and Proposed Solutions*, eds S. O. Lilienfeld and I. D. Waldman (Hoboken, NJ: Wiley Blackwell), 123–138.

Wasserstein, R. L., and Lazar, N. A. (2016). The ASA statement on p-values: context, process, and purpose. *Am. Stat.* 70, 129–133. doi: 10.1080/00031305.2016.1154108

Wellek, S. (2010). *Testing Statistical Hypotheses of Equivalence and Noninferiority*, 2nd Edn. Milton Park: Taylor & Francis.

Westfall, P., Johnson, W. O., and Utts, J. M. (1997). A Bayesian perspective on the Bonferroni adjustment. *Biometrika* 84, 419–427. doi: 10.1093/biomet/84.2.419

Westlake, W. J. (1972). Use of confidence intervals in analysis of comparative bioavailability trials. *J. Pharm. Sci.* 61, 1340–1341. doi: 10.1002/jps.2600610845

Woo, C. W., Krishnan, A., and Wager, T. D. (2014). Cluster-extent based thresholding in fMRI analyses: pitfalls and recommendations. *Neuroimage* 91, 412–419. doi: 10.1016/j.neuroimage.2013.12.058

Woolrich, M. W., Behrens, T. E., Beckmann, C. F., Jenkinson, M., and Smith, S. M. (2004). Multilevel linear modelling for FMRI group analysis using Bayesian inference. *Neuroimage* 21, 1732–1747. doi: 10.1016/j.neuroimage.2003.12.023

Woolrich, M. W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., et al. (2009). Bayesian analysis of neuroimaging data in FSL. *Neuroimage* 45, S173–S186. doi: 10.1016/j.neuroimage.2008.10.055

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

Conflict of Interest: 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Neuroinformatics

TECHNOLOGY AND CODE
published: 02 December 2021
doi: 10.3389/fninf.2021.720229

A Toolbox and Crowdsourcing Platform for Automatic Labeling of Independent Components in Electroencephalography

Gurgen Soghoyan[1,2], Alexander Ledovsky[2,3], Maxim Nekrashevich[3], Olga Martynova[2], Irina Polikanova[4], Galina Portnova[2], Anna Rebreikina[2], Olga Sysoeva[2] and Maxim Sharaev[2,3]*

[1] Center for Bioelectric Interfaces, National Research University Higher School of Economics, Moscow, Russia, [2] Laboratory of Human Higher Nervous Activity, Institute of Higher Nervous Activity and Neurophysiology, Russian Academy of Sciences, Moscow, Russia, [3] Research Center in AI, Skolkovo Institute of Science and Technology, Moscow, Russia, [4] Faculty of Biology and Biotechnology, National Research University Higher School of Economics, Moscow, Russia

OPEN ACCESS

Edited by:
Dezhe Z. Jin, The Pennsylvania State University (PSU), United States

Reviewed by:
Sheng-Hsiou Hsu, University of California, San Diego, United States
Tom A. Campbell, University of Helsinki, Finland
Alexander E. Hramov, Innopolis University, Russia

***Correspondence:**
Gurgen Soghoyan
gsogoyan98@gmail.com

Received: *03 June 2021*
Accepted: *03 November 2021*
Published: *02 December 2021*

Citation:
Soghoyan G, Ledovsky A, Nekrashevich M, Martynova O, Polikanova I, Portnova G, Rebreikina A, Sysoeva O and Sharaev M (2021) A Toolbox and Crowdsourcing Platform for Automatic Labeling of Independent Components in Electroencephalography. Front. Neuroinform. 15:720229. doi: 10.3389/fninf.2021.720229

Independent Component Analysis (ICA) is a conventional approach to exclude non-brain signals such as eye movements and muscle artifacts from electroencephalography (EEG). A rejection of independent components (ICs) is usually performed in semiautomatic mode and requires experts' involvement. As also revealed by our study, experts' opinions about the nature of a component often disagree, highlighting the need to develop a robust and sustainable automatic system for EEG ICs classification. The current article presents a toolbox and crowdsourcing platform for Automatic Labeling of Independent Components in Electroencephalography (ALICE) available via link http://alice.adase.org/. The ALICE toolbox aims to build a sustainable algorithm to remove artifacts and find specific patterns in EEG signals using ICA decomposition based on accumulated experts' knowledge. The difference from previous toolboxes is that the ALICE project will accumulate different benchmarks based on crowdsourced visual labeling of ICs collected from publicly available and in-house EEG recordings. The choice of labeling is based on the estimation of IC time-series, IC amplitude topography, and spectral power distribution. The platform allows supervised machine learning (ML) model training and re-training on available data subsamples for better performance in specific tasks (i.e., movement artifact detection in healthy or autistic children). Also, current research implements the novel strategy for consentient labeling of ICs by several experts. The provided baseline model could detect noisy IC and components related to the functional brain oscillations such as alpha and mu rhythm. The ALICE project implies the creation and constant replenishment of the IC database, which will improve ML algorithms for automatic labeling and extraction of non-brain signals from EEG. The toolbox and current dataset are open-source and freely available to the researcher community.

Keywords: EEG, automatic preprocessing, ICA, children, automatic artifact detection, machine learning algorithms

INTRODUCTION

Electroencephalography (EEG) signal reflects the bioelectrical activity of brain neuronal networks. For more than a century, human neuroscience and clinical research applied scalp EEG recording to study and assess a broad scope of sensory and cognitive functions. One of the crucial steps of EEG preprocessing is "purifying" the brain signal by extraction of the electrical activity of non-neuronal origins such as eye movements and muscle artifacts. For recent decades, Independent Component Analysis (ICA) offered a solution to this problem based on the isolation of statistically independent sources called independent components (ICs) as linear combinations of signals from electrodes (Makeig et al., 1996; Delorme and Makeig, 2004). A source of each IC can be projected onto the electrode cap and estimated via timecourse and spectral power. For example, ICA allows identifying components related to eye-movement and muscle artifacts based on their bioelectrical signals' specific characteristics, e.g., frequency and spatial distribution (Chaumon et al., 2015; Frølich et al., 2015). However, due to other frequent contaminations of EEG, a rejection of non-brain ICs is usually performed in the semiautomatic mode under the visual inspection of researchers. Herewith, labelings of ICs by different experts can substantially disagree, which might considerably affect the further analysis and reproducibility of EEG results (Robbins et al., 2020). Artifact rejection by ICA in children and patient EEG is especially challenging even for experts. The dependence of EEG analysis from subjective opinions of experts may explain that EEG data have been rarely included in large-scale studies or meta-analyses. For this reason, the automatic algorithms for EEG processing are the main objectives of many research groups (Nolan et al., 2010; Mognon et al., 2011; Winkler et al., 2011; da Cruz et al., 2018; Tamburro et al., 2018; Pedroni et al., 2019).

To create a robust and sustainable automatic system for EEG ICs classification, one needs to extract the most informative features from ICs and have an appropriate machine learning (ML) model inside the system. The accurate labeling of ICs is the crucial step in training and validating this model. The training of ML algorithms to automatically identify artifactual ICs will allow to set up a more objective methodology for EEG preprocessing.

Currently, a limited number of projects aims to create an automatic cleaning system of the EEG signal. For example, Automatic EEG artifact Detection based on the Joint Use of Spatial and Temporal features (ADJUST) (Mognon et al., 2011) and Fully Automated Statistical Thresholding for EEG artifact Rejection (FASTER) (Nolan et al., 2010) use empirical threshold-based algorithms. Machine learning approach was introduced in Multiple Artifact Rejection Algorithm (MARA) (Winkler et al., 2011), algorithms from the studies of Frølich et al. (2015) and Tamburro et al. (2018). SASICA software (Chaumon et al., 2015) is an EEGLAB Matlab plug-in (Chaumon et al., 2015), includes ADJUST, MARA, FASTER, and some other methods. The more novel study describes Adjusted-ADJUST approach (Leach et al., 2020) that is known as an advanced version for the previously described ADJUST software. It is aiming to produce automatic labeling for the pediatric ICA that differs from the ICA of adults because of infant EEG features. The suggested approach shows the higher quality even for adult data. All these studies used their private datasets for training and validation purposes. Those datasets were relatively small, consisting of several hundred ICs. In most cases, each IC was annotated by only one expert, which complicates the estimation of algorithm actual performance and comparison with other algorithms. Moreover, the lack of a large dataset with verified annotation limits the potential performance of machine learning models.

Pion-Tonachini et al. (2019) addressed this problem by proposing ICLabel Toolbox, which includes the annotation tool with crowdsourcing mechanics, datasets, and several machine learning algorithms. The annotation tool provides an interface to label a particular IC from the database by visualizing different components' characteristics. In this toolbox, the ML algorithms are based on artificial neural networks and claimed to be the fastest and most accurate than other studies.

While the ICLabel project is an excellent resource for automatic artifact rejection in EEG, it has several drawbacks. The first one is potentially insufficient annotation quality as a non-expert user can annotate ICs. It means that even if an ML algorithm has high accuracy, the predicted classes may be wrong as ICs have no order or intrinsic interpretations and their classification by experts requires practice. Potential technical issues that prevent the best performance from experts are inability to see other ICs from the same EEG record, which is helpful in ambiguous cases (e.g., horizontal eyes component can consist of two ICs, so seeing them in parallel helps to infer their nature) and limitation of component time-window plots to only 3 s ranges. Clinical experts usually require at least 30 s to properly detect various slow-wave components or alpha rhythm, hardly detected in a short time interval. Another limitation of ICLabel that the authors themselves pointed to is a limited variety of EEG data (Pion-Tonachini et al., 2019), as their dataset does not contain data from infants and most clinical groups.

The current study presents a toolbox and crowdsourcing platform for Automatic Labeling of Independent Components in Electroencephalography (ALICE), which is available via link http://alice.adase.org/. The ALICE toolbox aims to build a sustainable algorithm to remove artifacts and find specific patterns in EEG signals using ICA decomposition. The presented toolbox was also designed to overcome the limitations of the previous approaches mentioned above.

For developing a sustainable ML-based EEG component classification, the proposed toolbox should have two components: a high-quality labeled dataset of ICs and a proper ML pipeline to train and validate models.

Thus, the first aim of the ALICE project was to create a high-quality dataset with IC labels. In order to achieve this goal, we performed the following steps:

- The definition of a rigorous set of possible IC classes that would cover a wide variety of cases and be easily understandable by experts.
- The annotation of IC reliability by combining opinions from multiple experts.

- Resolving the possible poor concordance between experts by various merging strategies.
- Attracting researchers to share their datasets, including unique EEG recordings from rare clinical groups.

The second aim of the ALICE project was to develop a robust but flexible ML pipeline for automated IC classification. The ML module includes implementing various features (both well-established and original), multiple ML models, and the validation pipeline. The ALICE project also invites the research community to develop their models using our dataset, which is available via link http://alice.adase.org/downloads.

The other ambitious goal for ALICE development is the automatic identification of components related to the functional brain oscillations, such as alpha and mu rhythm. Mu rhythm overlaps with alpha rhythm in a frequency range of 8-13 Hz but has a different oscillation shape and localization at scalp electrodes. While alpha rhythm is recorded predominantly from the occipital lobes with closed eyes and suppressed when eyes open (Berger, 1931), mu rhythm emerges over the sensorimotor cortex and is attenuated during movements. Importantly, mu rhythm does not react to opening or closing the eyes (Kuhlman, 1978). Despite the described differences, the automatic separation of mu from alpha waves in EEG is challenging and drawing the attention of many methodological studies (Cuellar and del Toro, 2017; Garakh et al., 2020). Still, the identification of mu rhythm often requires visual inspection and expertise. The ALICE toolbox aims to accumulate expert labeling of alpha and mu rhythms to improve automatic identification of functional brain oscillations by supervised ML.

MATERIALS AND METHODS

Automatic Labeling of Independent Components in Electroencephalography Toolbox High-Level Architecture

Automatic Labeling of Independent Components in Electroencephalography contains two modules (**Figure 1**):

- Annotation module, which consists of a user interface (UI) and ICs database. An HTTP API allows uploading ICs data to the database. Web-based UI allows experts to label uploaded data for future ML models training and validation.
- ML module is based on a Python library, which trains ML models based on expert annotations and uses pre-trained ML models to apply to new IC data.

Annotation Module

By annotation, we mean a process of manual IC labeling by experts based on various data visualization tools available at the ALICE platform, such as IC topographic mapping, plots of time series, and power spectrum. An expert may choose IC labels from a predefined number of options.

We propose a set of IC component labels including major artifact types with subtypes as well as brain signal subtypes:

- Eye artifacts – eye movement artifacts of any type.
- Horizontal eye movements – components that represent activity during eye movements in horizontal directions.
- Vertical eye movements – components that represent activity during eye movements in vertical directions.
- Line noise – line current noise evoked by surrounding electrical devices.
- Channel noise – the noise associated with channels that can be Or.
- Brain activity – brain activity of any type.
- Alpha activity – alpha rhythm with oscillation in the frequency band of 8–13 Hz with predominance in the occipital lobe channels.
- Mu activity – mu rhythm with oscillation in the frequency band of 8–13 Hz with predominance or dipole localization in the frontal-central-parietal area.
- Muscle activity – artifacts from a recording of muscle activity on the head surface.
- Heartbeat artifacts – artifacts that represent electrocardiographic activity.
- Other – components with explicit nature that label is not listed in the labeling system, for example, breathing (experts could comment on the label choice in the comments section, the ALICE developers collect data from comments and expand the list of labels in the subsequent versions of the toolbox).
- Uncertain – components with unclear nature.

The web-based UI supports the annotation process (**Figure 2**). An expert has the following data visualization options:

- Topomap of IC weights.
- Power spectrum density plot.
- Plot of all ICs time series for the current subject (the time-series length is 30 s with the possibility of scrolling and zooming selected time interval).
- Epoch image illustrates the color-coded amplitude fluctuations of the IC in all available EEG epochs and averaged ICs time series values.
- This plot is helpful for the annotation of epoched data.

After a particular expert has finished the labeling process, the data of ICs with annotations can be packed into an archive by the annotation module by an administrator. Then, annotated data becomes available at the Downloads page of the ALICE toolbox and could be used both by the experts and ALICE data scientists.

Machine Learning Pipeline

There could be many discrepancies between experts' annotations due to ambiguities in IC patterns, data quality, and differences at the expert level. The annotation inconsistency means that we need to create final IC labels in the dataset as a function of the individual annotations. So, before conventional ML pipeline steps, such as *Feature calculation* and *ML model training and selection*, we need to include an additional step – *Data label aggregation*. The whole data processing and ML pipeline are presented in **Figure 3**, and each step is discussed in detail below.

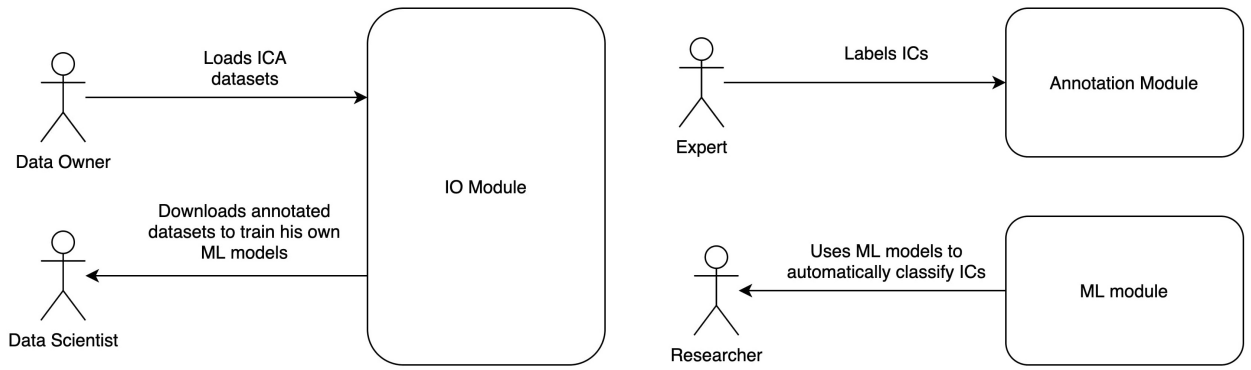


FIGURE 1 | ALICE toolbox high-level architecture and user roles. Annotation module UI serves for Data Owners to upload IC data to the database and for Experts to provide annotation on existing or newly uploaded data. Data Scientists and Researchers work with ML module: the former train models based on selected samples from the database, the latter take pretrained models to work with their own data (online or offline). Online version of the ALICE toolbox is available at http://alice.adase.org/.

Data Labels Aggregation

This part aims to create a boolean variable between each component and each IC class, reflecting whether a specific activity is present or not in a particular component. The first step is to create an annotation table (**Figure 4A**). *The annotation* is a term denoting the labeling produced by an expert to a particular component. Experts have their own unique opinion about the component's ICA class. Our goal is to develop an approach to grouping expert annotations to form a common opinion on each component.

A simple voting strategy seems to be a logically correct option: if most experts choose that a component contains a particular activity, for example, an eye artifact, then this component is classified as an eye artifact. This approach is the basis of Strategy 1, which we called "*Majority vote*," although it does not require that the majority (more than 50%) of the experts assign the component this particular label. The threshold value can be changed. We provided an example where it equals to 33% which means we expect agreement over 33% of experts. In other words, by grouping experts' annotations, we form the average of the experts' votes (**Figure 4C**). We will consider this average value as the *probability* of assigning the component to a specific class. If the probability is higher than the threshold, we assume that the component encodes the given IC class; otherwise, it does not (**Figure 4D**).

Nevertheless, if an expert assigned a component to several classes, it means s/he recognizes several types of activity present in the IC. This situation can lead to ambiguous results if the expert acted with an approach where s/he labels mixed components with all types of activity s/he believes are potentially intermixed in a particular IC. If we were to use *Majority vote* for such situations, it would lead to low quality of the target variable as IC with only one label is a more genuine representation of this class than the component that contains a mixture of artificial and brain activity. An example of what this can affect is illustrated in **Figure 5**. We see that the component, due to such markup, is assigned to all classes simultaneously.

In order to overcome this situation, Strategy 2 was developed and titled "Probabilistic vote." Imagine that, when labeling a component, an expert has one vote, which they equally distributed among all the classes to which they attributed this component. In other words, if a person marks a component as eyes and as muscles, and as heart, then with a *probability* of 0.33, they assign it to each of these classes (**Figure 4B**). Further, these *probabilities* are again averaged (**Figure 4C**). Then, a threshold is chosen, according to which it is decided whether this *weighted probability* will be transformed to 1 or 0 (**Figure 4D**). The threshold of 0.33 was chosen as the optimal threshold for the current data, assuming that components that consist of three or fewer labels still represent the simple pattern of interest for the model. This approach is rather valuable for cases where the mixed nature of components can affect the target variable; **Figure 5** provides the example.

The threshold value is highly dependent on the level of agreement between experts since a too tight threshold with a low agreement will significantly reduce the number of objects. On the other hand, a weak threshold with a high agreement will lead to noisy, ambiguous components in the training set. We decided to use an equal threshold of 0.33 for both strategies. The threshold change for *Majority vote* will make sense with an increase in the number of experts.

Agreement Between Experts

We also computed metrics of expert agreement to be able to compare annotation quality of various classes as well as datasets. For the case of two experts, we propose using Cohen's kappa (Cohen, 1960).

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

Where p_0 is the relative observed agreement (similar to accuracy), p_e is the hypothetical probability of agreement by chance.

For the case of multiple experts, we propose using Fleiss' kappa (Fleiss and Cohen, 1973), which has a similar

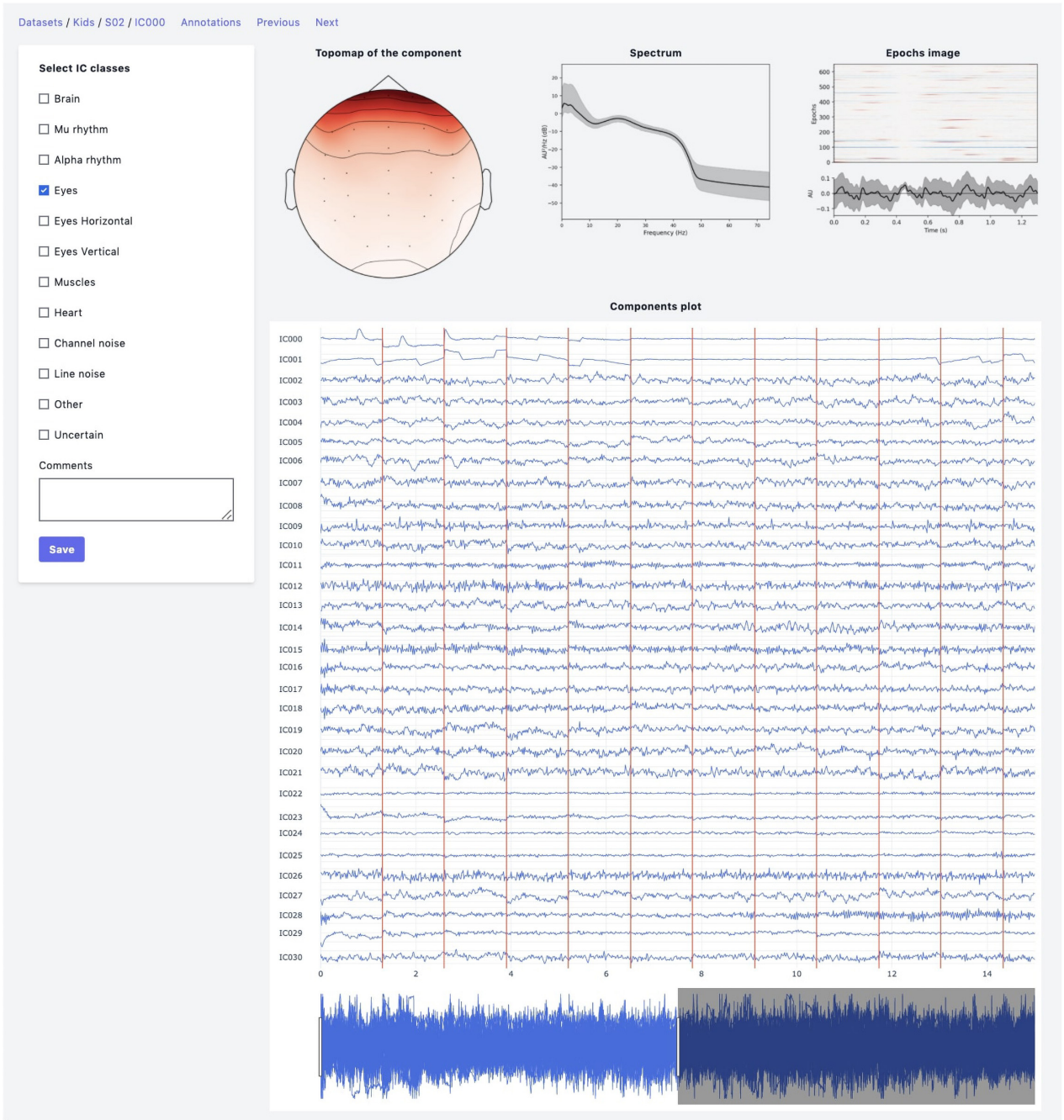


FIGURE 2 | Annotation module interface. Top row: IC topomap and spectrum as well as Epochs image, illustrating the color-coded amplitude fluctuations (arbitrary units) of the IC in all available epochs (time relative to sound presentation on the *x*-axis, epochs on the *y*-axis). Image at the bottom shows averaged values of ICs timeseries; bottom row: all IC for considered subject are plotted together.

formula with a different definition of p_0 and p_e, that depend on weighted estimates. Basically, that shows the level agreement between the multiple experts above the value of agreement expected by chance for details refer to Fleiss and Cohen (1973).

Based on the metrics from Pion-Tonachini et al. (2019), we computed the inter-expert correlation between experts to compare our level of agreement with the level of agreement in ICLabel.

$$IEC = \frac{1}{N}\sum_{n=1}^{N} Corr(v_{1,n}, v_{2,n})$$

N, number of components marked by both experts; $v_{1,n}$, annotation vector made by the 1st expert corresponding to the

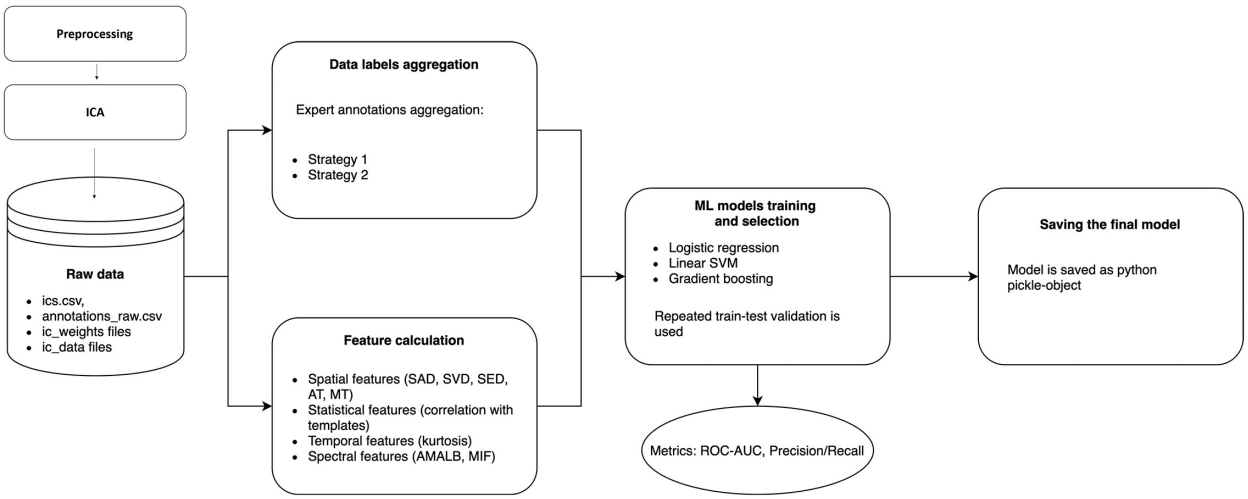


FIGURE 3 | Data processing and machine learning pipeline in ALICE. Raw ICs data with annotations are passed to Data labels aggregation and Feature calculation blocks to form a labeled dataset and extract informative features from ICs. Three ML models are trained with repeated train-test validation and different model quality metrics are calculated. The best model is then selected and could be exported as a Python pickle-object.

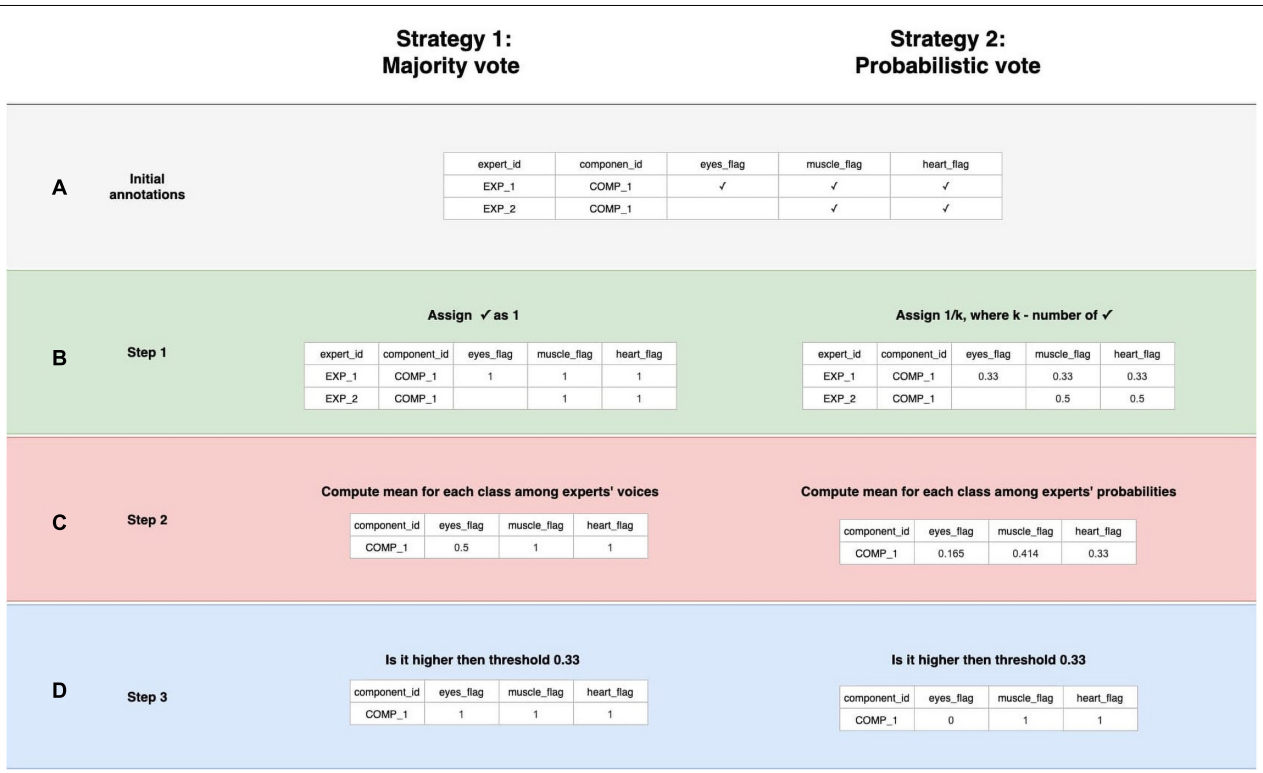
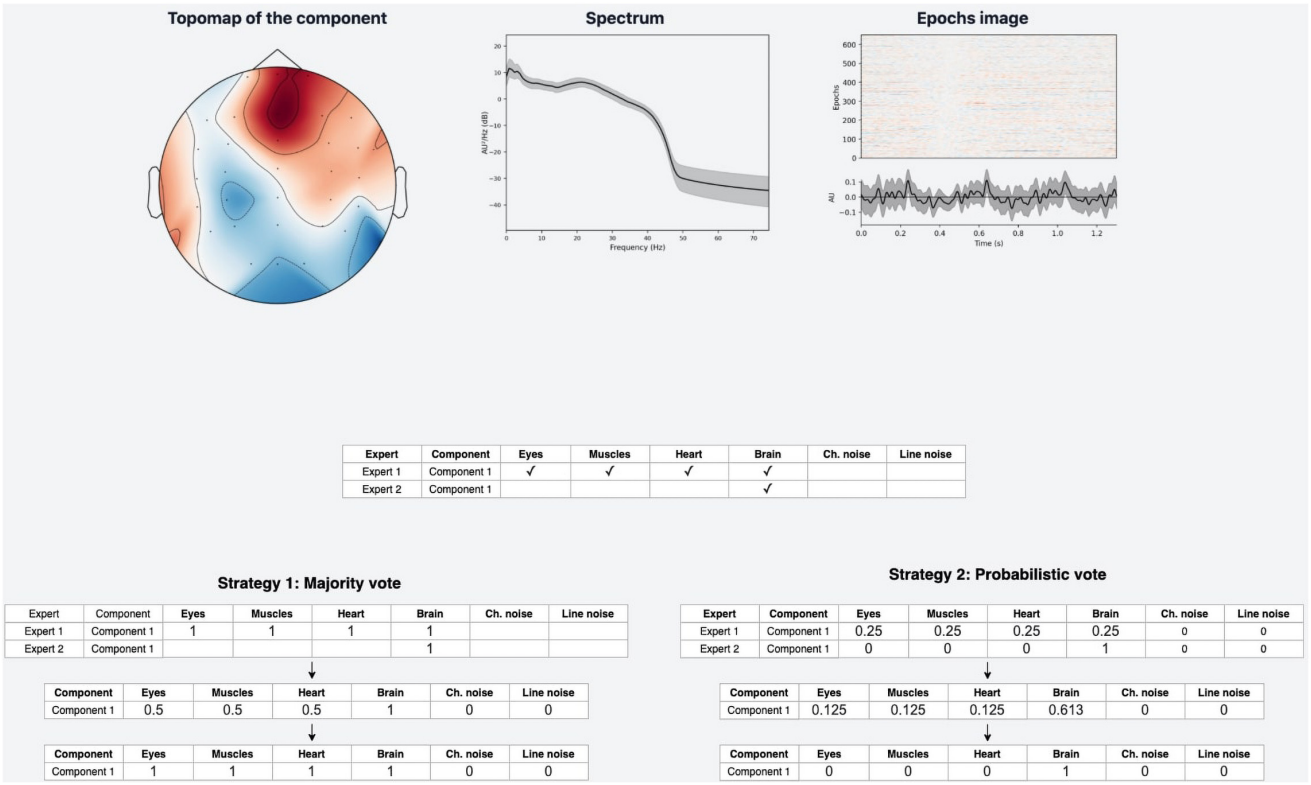


FIGURE 4 | Data labeling strategies. There are several annotations for one component from various experts, but we strive to designate its belonging to a particular class strategies of data belling aggregation, those are "Majority vote" and "Probabilistic vote." **(A)** Is a table of annotations of a specific component. **(B)** Transform into a matrix of *probabilities* that each component belongs to a particular class. **(C)** Group the experts' opinions using the mean of the *probabilities* to obtain a *weighted probability*. **(D)** Determine whether the weighted probability is higher than the threshold or not.



Expert	Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Expert 1	Component 1	✓	✓	✓	✓		
Expert 2	Component 1				✓		

Expert	Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Expert 1	Component 1	1	1	1	1		
Expert 2	Component 1				1		

Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Component 1	0.5	0.5	0.5	1	0	0

Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Component 1	1	1	1	1	0	0

Expert	Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Expert 1	Component 1	0.25	0.25	0.25	0.25	0	0
Expert 2	Component 1	0	0	0	1	0	0

Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Component 1	0.125	0.125	0.125	0.613	0	0

Component	Eyes	Muscles	Heart	Brain	Ch. noise	Line noise
Component 1	0	0	0	1	0	0

FIGURE 5 | Discrepancies in experts' opinions and difference in strategies handling. Experts may have different approaches in labeling samples with mixed nature of activity. Expert 1 marked such components with all activity types that are present in the component. Expert 2 instead focused on the components that have a clear pattern of a single IC class. Within a provided example we can see how the difference in their approaches lead to different annotations for the same IC component. The application of "Majority vote" and "Probabilistic vote" strategies for the suggested example affect significantly the final label of the component. With the first one the component corresponds to many IC classes, while in the second case the component is assigned only as a Brain activity.

n^{th} component; $v_{2,n}$, annotation vector made by the 2nd expert corresponding to the n^{th} component.

All computational details about data label aggregation are available via link https://github.com/ledovsky/alice-eeg-ml that we share with interested researchers who might achieve higher performance rates on our dataset using their settings for strategies and thresholds.

Features Calculation

To reduce data dimensionality while preserving the most characteristic information for each IC class, we calculate specific temporal and spatial features of each signal. Most features are well established and based on previous research. Still, we introduced some modifications to existing ones and treated them as new features.

Among the established features are:

- Kurtosis of the component time series (Nolan et al., 2010; Mognon et al., 2011; Winkler et al., 2011; Tamburro et al., 2018). By definition, kurtosis is the fourth standardized moment of the time series. In epoched data, we calculate an average of the feature computed for each epoch separately. It helps to distinguish ICs that correspond to eyes and brain activity.
- Maximum Epoch Variance (Mognon et al., 2011; Tamburro et al., 2018) is used to detect eye movements. The value of this feature is a ratio between the maximum of the signal variance over epochs and the mean signal variance. As proposed in Mognon et al. (2011), we excluded one percent of the largest variance values to improve its robustness when calculating this feature.
- Spatial Average Difference (SAD), Spatial Variance Difference (SVD), and Spatial Eye Difference (SED). Spatial features proposed in Mognon et al. (2011) depend on IC weights of eyes-related electrodes. SAD is calculated as the difference between channel weight averages over frontal and posterior regions. SVD is the difference between weight variances in these regions. These are used to distinguish vertical eye movements. SED is the difference between the absolute values of weight averages in the left eye and right areas. This feature detects horizontal eye movements.
- Myogenic identification feature (MIF) (Tamburro et al., 2018) is used to detect muscle activity and is calculated as the relative strength of the signal in the 20–100 Hz band.

- Correlations with manually selected signal patterns (Tamburro et al., 2018). We use these to detect eye blinks and eye movements.

The ALICE toolbox also offers a possibility of mu and alpha rhythms annotation and classification. Thus, some features must be specific to these components' spatial and temporal properties.

Alpha rhythm is known to be localized in occipital and parietal areas with increased power in 8–12 Hz for adults. Close to the alpha band frequency, mu rhythm is generated in central and frontal areas. We used those electrodes that maximally emphasize the contrast between mu and alpha localization by the topography-related features. Thus, the original features include:

- Mu topography (MT): A feature which is sensitive to topomaps of mu rhythm ICs, where *Mu* is the following set of electrodes in 10–20: "Fp1," "Fpz," "Fp2," "F3," "Fz," "F4," "Fc3," "Fcz," "Fc4," "C3," "Cz," "C4."

$$MT = \sum_{e \in Mu} |w_e| - \sum_{e \notin Mu} |w_e|$$

- Alpha topography (AT): A feature which is sensitive to topomaps of alpha rhythm ICs where *A* is the following set of electrodes in 10–20: "C3," "Cz," "C4," "Cp3," "Cpz," "Cp4," "P3," "Pz," "P4," "O1," "Oz," "O2."

$$AT = \sum_{e \in A} |w_e| - \sum_{e \notin A} |w_e|$$

- Average magnitude in alpha band (AMALB): The ratio between average amplitude in the alpha band (6–12 Hz) and average amplitude in other frequencies (0–6 Hz; 13–125 Hz) is sensitive to alpha ICs. The alpha range was expended to 6 Hz because alpha band tends to be in the lower frequency range for children (Marshall et al., 2002; Lyakso et al., 2020).

$$AMALB = \frac{\sum_{f \in [6,\ 12]} \underline{x}(f)}{\sum_{f \notin [6,\ 12]} \underline{x}(f)}$$

Source code used to compute the features can be found via link https://github.com/ledovsky/alice-eeg-ml.

Machine Learning Models Training and Selection

The current version of ALICE Toolbox provides three different machine learning models: logistic regression (LR), linear support vector machine (SVM), and gradient boosting (XGB). These models are built on different principles and are relatively simple compared to neural networks and deep neural networks. Keeping in mind a relatively small initial dataset, we considered the three models mentioned above as an optimal initial model choice. All of them are optionally available for new training and testing procedures in ALICE. In particular, we used the LR implementation from scikit-learn package (Pedregosa et al., 2011) with default parameters (including regularization parameter C = 1.0, L2 penalty and liblinear solver). Linear SVM is taken from scikit-learn package (Pedregosa et al., 2011) with default parameters (including regularization parameter C = 1.0). Finally, we used the XGB model implementation from XGBoost package (Chen and Guestrin, 2016) with default patameters of 30 estimators with a maximal depth of 4.

In the ALICE, we implement the repeated train-test split cross-validation technique. We trained the model on 70% of samples and validated on the rest 30% with repeated train-test cross-validation and did not optimize any hyperparameters on cross-validation. We performed this procedure 50 times using different random train-test splits, estimating three main metrics of classification accuracy: Area Under the Receiver Operating Characteristic Curve (ROC-AUC), Area Under the Precision-Recall Curve (PR-AUC) and F1-score using the implementation of scikit-learn package (Pedregosa et al., 2011). ROC-AUC and PR-AUC were used as overall metrics of model performance for different thresholds and considered the main ones. F1 was used as a performance metric of optimal model splits and was considered as an additional metric.

Thorough code used for computations is open access https://github.com/ledovsky/alice-eeg-ml/blob/main/Basic%20Pipeline.ipynb. Thus, any person can go through our pipeline and make his/her changes to achieve higher results and easily compare them with our original performance rates. The Basic Pipeline explains how the models may be applied to any dataset.

Initial Dataset

The ALICE project aims to involve the neurophysiological community in labeling existing publicly available and new IC datasets to improve ML models' quality. However, the Baseline model trained on the dataset provided by IHNA&NPh RAS is already available to users.

Electroencephalography data were recorded using the NeuroTravel amplifier (EB Neuro, Italy) with sampling rate 500 Hz, and with 31-scalp electrodes arranged according to the international 10–10 system and included the following electrodes: "Fp1," "Fpz," "Fp2," "F3," "Fz," "F4," "F7," "F8," "FC3," "FCz," "FC4," "FT7," "FT8," "C3," "Cz," "C4," "CP3," "CPz," "CP4," "P3," "Pz," "P4," "TP8," "TP7," "T3," "T4," "T5," "T6," "O1," "Oz," "O2." Ear lobe electrodes were used as reference, and the grounding electrode was placed centrally on the forehead. The initial dataset consists of recordings from 20 typically developing children aged 5–14 years. Within the experiment's framework, sound stimulation was performed according to the odd-ball paradigm with a standard stimulus of 1,000 Hz and two deviant stimuli at 980 and 1,020 Hz. The interstimulus interval was 400 ms. Stimulus intensity were 75 dB.

Obtained data were filtered (0.1–40 Hz) and divided into epochs (−500; 800 s), where noisy epochs were removed by threshold (350 mV). Only the first 650 epochs of recording gained from the first 650 presentations of stimuli were used for posterior ICA decomposition (FASTICA) with resampling on the level of 250 Hz. Final data that were uploaded into ALICE consisted of 30 ICA components. All preprocessing steps were done using the MNE Python package (Gramfort, 2013).

The data annotation for training the Baseline model was carried out by two experts – experienced scientists of the Institute of Higher Nervous Activity and Neurophysiology of RAS. The first expert is a clinical neurologist, while the second one clinical psychologist; both experts had more that 15 years of experience

in analysis of pediatric EEG ICA. For the correct work with the program, they received an instruction, which outlined the main steps they took when working with ALICE. Experts' main task was set as follows – to mark each component using the set of labels: Eyes, Horizontal eye movements, Vertical eye movements, Line noise, Channel noise, Brain, Alpha activity, Mu activity, Muscle activity, Other, Uncertain. Following the instructions, if an expert saw that a component consisted of several activity types, s/he can assign the component to several classes. For example, among annotated components, there were often components marked both as eye artifacts and muscle activity simultaneously.

Additional Datasets

For additional validation we used another dataset with children EEG. The recordings of 17 children aged 5–14 years were decomposed using ICA. Data were recorded using the same EEG system as in initial dataset. Participants watched series of videos in terms of the experimental paradigm. The collected data were filtered in the range 3–40 Hz, no other processing steps were applied. The same experts were asked to mark only those components that correspond to eye artifacts. The overall number of components was 149. The dataset is marked as Children dataset 2.

To test how ALICE performs on adult data the recordings of 21 adults were added to the ALICE platform. The experimental design, EEG system and data processing steps were the same as we used in the initial dataset. The data were annotated by four new experts. To facilitate the labeling process, the task for experts was to label only those components that correspond to eye artifacts. The datasets is called Adults' dataset.

These additional datasets allowed us to estimate model performance when trained using initial dataset and re-trained on additional datasets. Moreover, it was mentioned previously that adult ICA and children ICA automatic labeling require different approaches to modeling. Thus, the second dataset allowed us to check whether suggested approach is suitable for EEG of any age, whereas the first dataset was acquired from the same cohort of participants. In order to assess model generalizability, data preprocessing was also different: the first dataset was prepared with different ICA method – AMICA (Palmer et al., 2011).

Ethics Statement

The datasets were obtained from the research project (A physiological profile of autism spectrum disorders: a study of brain rhythms and auditory evoked potentials). It is conducted according to the guidelines of the Declaration of Helsinki and approved by the Ethics Committee of the Institute of Higher Nervous Activity and Neurophysiology (protocol code 3, date of approval July 10, 2020). All children provided their verbal consent to participate in the study and were informed about their right to withdraw from the study at any time during the testing. Written informed consent was also obtained from a parent/guardian of each child.

RESULTS

Data Labeling Aggregation

First, we explored the level of consistency between two annotators for various IC classes. Due to limited available data and only two annotators, we decided to merge some classes with a small number of label matches between annotators. One reason for this small number could be the possible difference in labeling strategies between the experts, as was discussed in the section "Materials and Methods." The final manipulations with class labels are:

- Eyes, Horizontal eye movement, Vertical eye movement were merged to the one Eye movement class.
- Line noise labels were dropped due to a lack of actual line noise in available data.
- Alpha and mu labels were checked to be marked as a Brain label too.

For the rest of the IC classes, we used the following aggregation strategies based on each class's total number of positive samples (see **Table 1**). When the samples of a particular class were poorly represented, we took *Majority vote strategy* to have enough labeled samples for the model fitting; otherwise, we took Probabilistic vote strategy. The details of *Majority vote* and Probabilistic vote are explained in the section "Materials and Methods."

The final number of positive labels and concordance between the two experts are shown in **Table 2**.

According to arbitrary settled thresholds (Landis and Koch, 1977), the agreement between two experts' opinions was highest but still moderate (<0.4) only for labeling the ICs of brain signals. The other ICs were labeled with a relatively poor agreement between experts (**Table 2**). The Inter-expert correlation between our experts equals 0.43, and the approximate level of agreement was also reviewed (Pion-Tonachini et al., 2019). Based on the experts' comments, we understood that many IC components contain more than one activity type. This mixture led to uncertainty for experts' labeling strategy. Summing up their annotations and based on the comments, we can conclude that one expert was inclined to label only those components where a clear pattern of chosen IC class could be detected. Another expert labeled all activity types present at given components, even when there was only a slight indication of its presence in multi-nature ICs. This difference in labeling strategies produced relatively poor agreement even for (usually well recognized) Eyes activity. The annotation dataset is available via http://alice.adase.org/downloads.

Independent Component Classification

As it can be seen from **Table 2**, many classes are relatively small. This leads to imbalanced classification tasks, for example, for Alpha, Mu, and Channel noise IC classes. In this case, Precision-Recall (PR) curve better reflects classifier performance compared to the conventional ROC-AUC curve. So, we explored LR, XGB, and SVM as ML models and calculated both ROC-AUC and PR-AUC scores as performance measures. We selected among

TABLE 1 | IC classes and corresponding aggregation strategies based on the total number of positive samples of each class.

IC class	Brain activity	Alpha brain activity	Mu brain activity	Eyes	Muscles	Heart	Channel noise
Strategy	2	1	1	2	2	1	2

TABLE 2 | Number of samples and Cohen's kappa for each class.

Label	Number of samples	Cohen's kappa
Brain	449	0.47
Alpha	60	0.13
Mu	92	0.22
Eyes	78	0.10
Muscles	135	0.36
Heart	231	0.04
Channel noise	48	0.12

three models for each IC type separately. All the models showed comparable performance for most ICs classes (see **Figure 6** for ROC curves and **Table 3** for values) based on ROC-AUC curves. Brain, Eyes, and Muscles models showed the best performance among others with ROC-AUC greater than 0.9. We could not train a good model for Heart ICs detection due to inadequate labeling as suggested by the lack of consistency among experts and probably not specific extracted features.

However, the picture was different when analyzing PR curves and PR-AUC values (see **Figure 7** and **Table 4**). As we mentioned, PR curves better indicate classification performance in case of imbalanced data, which results in worse performance for Alpha, Mu, and Eyes IC types, all of which have fewer positive labels than Brain or Muscles IC classes. It also can be seen that for Heart and Channel Noise classes, all of the models and SVM in particular performed poorly. The possible reasons for this might be both a small number of samples in each class and a low level of agreement between the annotators resulting in poor labeling quality and lack of robustness. Probably, new robust predictive features should be developed to address these types of artifacts. We also provided F1-score values (see **Table 5**), alternative statistics based on precision-recall interaction. The need for further investigation of the models' performance on Heart and Channel Noise IC classes is also backed up by the low F1-score, which is significantly lower than the rest IC types.

It is worth mentioning that the main reason for measuring PR-AUC was to compare the performance of the models with each other. In general, specific PR-AUC values, unlike ROC-AUC, do not reflect the model's performance. For that, it is better to refer to the PR curve itself. Each point on this curve corresponds to certain precision and recall levels closely related to type I and II errors, respectively. We could achieve this by choosing the appropriate threshold (by default, each model predicts probabilities for each class that can be interpreted as either True or False by comparing with the threshold value). To better illustrate this idea, we suggest the following example. Supposing, we want to detect muscles with the recall of 0.75 (that is, we will detect 75% ICs with muscular activities). Then, by looking at **Figure 7**, we can see that SVM will achieve a precision value of about 0.7, which means that out of all ICs selected, about 70% will correspond to Muscles.

We chose an ML model for each IC type based on the ROC-AUC score if the class is relatively balanced (Brain and Heart and

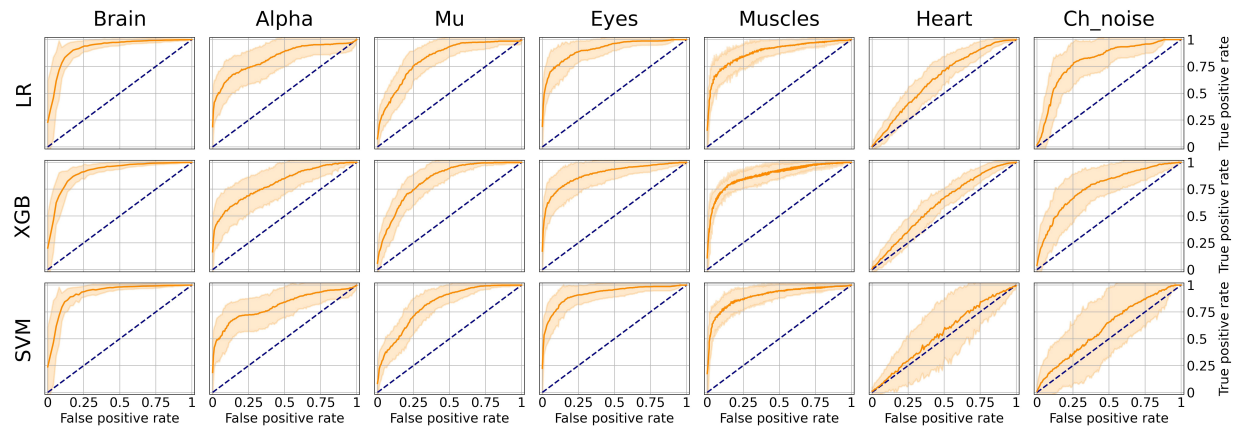


FIGURE 6 | Aggregated Receiver Operating Characteristic (ROC) curves for all IC types and ML models. The solid line indicates the mean curve and the colored area indicates the 95% confidence interval for the ROC curve. The best classification results were achieved for the Brain Muscles and Eyes ICs. For the Alpha Mu and Channel Noise classes, the scores are also high, however, the stability is lower, especially in the case of detecting Channel Noise using SVM. Finally, the performance on Heart components was poor, which could be due to low expert concordance. The blue line on each plot represents the no-skill classifier which assigns labels at random. Thus, we can consider the performance of a particular model on a particular label type statistically significant, if the confidence interval lies above the blue line. Thus, most of our models classify the components significantly better than at chance, expert for SVM that was not able to do it for Heart and Channel noise component.

TABLE 3 | Average ROC-AUC values and their standard deviations.

ROC-AUC	Brain	Alpha	Mu	Eyes	Muscles	Heart	Channel noise
Logistic regression	0.93 (±0.02)	0.83 (±0.05)	0.83 (±0.03)	0.91 (±0.03)	0.89 (±0.03)	0.64 (±0.04)	0.81 (±0.05)
XGBoost	0.92 (±0.02)	0.81 (±0.05)	0.83 (±0.03)	0.89 (±0.04)	0.88 (±0.02)	0.61 (±0.03)	0.77 (±0.05)
Support vector machine	0.93 (±0.02)	0.81 (±0.05)	0.82 (±0.03)	0.92 (±0.03)	0.90 (±0.03)	0.55 (±0.10)	0.61 (±0.09)

Mean ± St. deviation.

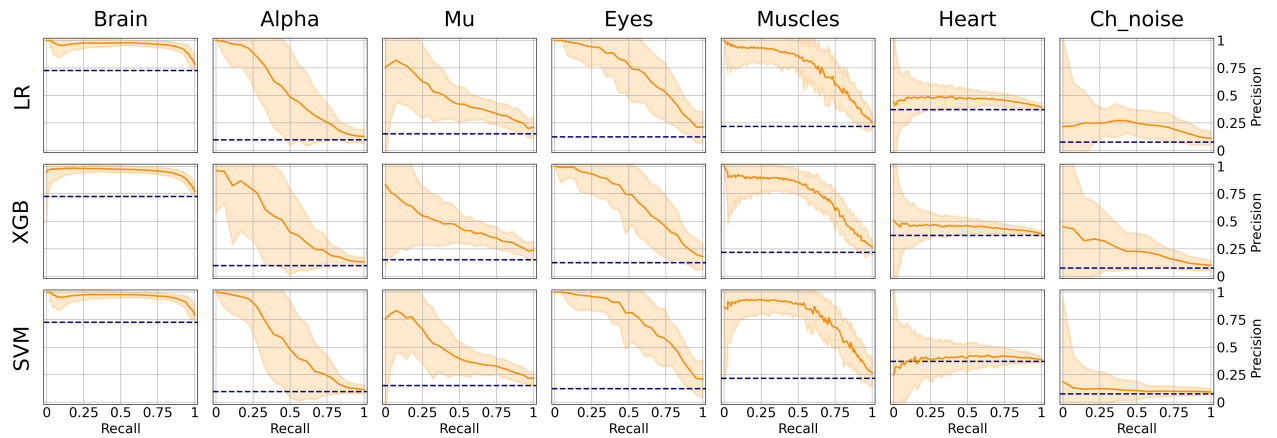


FIGURE 7 | Aggregated Precision Recall (PR) curves for all IC types and ML models. The solid line indicates the mean curve and the colored area indicates the 95% confidence interval for the PR curve. PR curves better indicate classification performance in case of imbalanced data, which can be seen in worse results for Alpha, Mu, Eyes, and especially Channel Noise IC types, all of which have fewer positive labels compared to Brain or Muscles IC classes. As with the ROC curves, we can claim that on all IC types except for Heart and Channel Noise, our models perform significantly better than the unskilled classifier.

TABLE 4 | Average PR-AUC values and their standard deviations.

PR-AUC	Brain	Alpha	Mu	Eyes	Muscles	Heart	Channel noise
Logistic regression	0.96 (±0.02)	0.59 (±0.08)	0.50 (±0.07)	0.74 (±0.06)	0.77 (±0.05)	0.46 (±0.04)	0.23 (±0.06)
XGBoost	0.96 (±0.01)	0.54 (±0.10)	0.48 (±0.07)	0.71 (±0.07)	0.75 (±0.05)	0.45 (±0.03)	0.27 (±0.09)
Support vector machine	0.96 (±0.02)	0.59 (±0.08)	0.49 (±0.07)	0.76 (±0.06)	0.79 (±0.05)	0.41 (±0.07)	0.13 (±0.04)

Mean ± St. deviation.

TABLE 5 | Average F1-scores and their standard deviations.

PR-AUC	Brain	Alpha	Mu	Eyes	Muscles	Heart	Channel noise
Logistic regression	0.92 (±0.01)	0.50 (±0.11)	0.31 (±0.08)	0.62 (±0.08)	0.66 (±0.05)	0.14 (±0.05)	0.00 (±0.00)
XGBoost	0.91 (±0.01)	0.50 (±0.12)	0.39 (±0.08)	0.64 (±0.07)	0.69 (±0.04)	0.40 (±0.04)	0.18 (±0.11)
Support vector machine	0.92 (±0.01)	0.42 (±0.10)	0.20 (±0.09)	0.63 (±0.07)	0.72 (±0.04)	0.01 (±0.02)	0.00 (±0.00)

Mean ± St. deviation.

Muscles) and based on PR-AUC if the class is unbalanced. Thus, we selected PR for Brain, Alpha Mu, and Heart, XGB for Channel Noise, and SVM for Eyes and Muscles.

Additional Tests

The obtained models were applied to additional datasets to decode eye artifacts. The model trained on the main dataset showed a controversial result while being tested on Children dataset 2 (F1-score = 0.12; PR-AUC = 0.18; ROC-AUC = 0.5). Nonetheless, the models perform well after retraining (**Figure 8A**) with PR-AUC values on the level of 0.94 (see **Table 6**). The latter implies that model re-training is beneficial for new datasets (even of the same age cohort) making model flexibility an important part of the proposed framework. After aggregation of experts labeling the dataset consisted of 64 eye components out of 527.

We also tested model performance on eye components from Adults' dataset (with 61 labeled eye-components out of 337). The model trained on initial dataset again performed weakly (F1 = 0.36; ROC-AUC = 0.61; PR-AUC = 0.65), while the re-trained models showed a dramatic increase in all quality measures (see **Figure 8B**). The obtained performance rate makes up 0.95 for PR-AUC. Thus, we examined that ALICE machine learning pipeline is also appropriate for adult EEG.

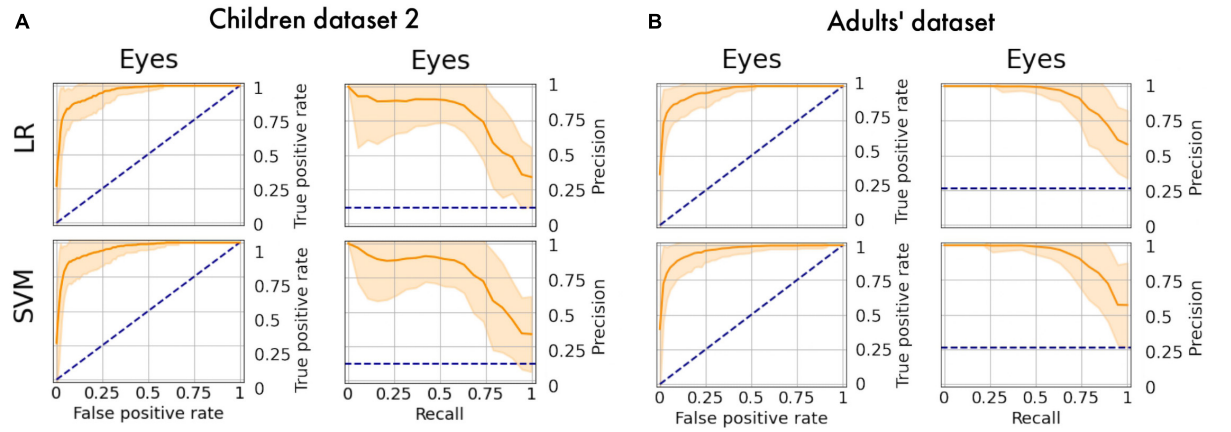


FIGURE 8 | PR curves and ROC curves for Eyes class for additional datasets. The performance rate for additional datasets illustrated using ROC-curve and PR-curve. The solid line indicates the mean curve and the colored area indicates the 95% confidence interval for the curves. **(A)** Is a pair of plots for Children dataset 2 and we observe that both plots illustrate high quality of predictions for Eye components. **(B)** Is a pair of plots that show a high performance rate for Eye components for Adults' dataset.

TABLE 6 | Performance rate on additional datasets.

	Children dataset 2			Adults' dataset		
	ROC-AUC	PR-AUC	F1-score	ROC-AUC	PR-AUC	F1-score
Logistic regression	0.96	0.81	0.76	0.97	0.94	0.83
Support vector machine	0.96	0.81	0.76	0.97	0.94	0.83

DISCUSSION

Independent component analysis is a powerful tool for the segregation of various types of activities from the raw EEG data. It is widely used for the detection of different artifacts such as eye blinks or muscle contractions. Nevertheless, IC signals' correspondence to any class of activity largely depends on a particular expert, affecting the study results. This issue is worth highlighting as the application of ICA in EEG studies becomes more and more popular. The ALICE toolbox is a particular instrument to resolve these issues.

The developed web application stores ICA data and makes it publicly available. This data includes IC annotations given by experts, which assign each component to the appropriate category. Moreover, the annotated dataset expands using the interface where each expert can make their labeling. ALICE's goal is to build a community where experts from neuroscience, neurophysiology, and other related areas, share their ICA data and encourage each other to make the annotations. Our study's low Cohen's kappa coefficient and low inter-expert correlation in IC annotation point to high disagreement in components annotation evident even between two experts. Noteworthy, the only other crowdsourcing platform for IC classification [ICLabel, (Pion-Tonachini et al., 2019)] also report similar results: their mean inter-expert correlation was 0.50, ranging from 0.46 to 0.65, clearly pointing to different strategies of identification ICs. This finding emphasizes the need to study the reason for such low agreement between experts and to develop an automatic IC classification toolbox that will work objectively.

The ALICE has the potential to unite the efforts of experts from different fields that are vital to developing an ML model that could be used in EEG studies for the objective assessment of various artifacts. Our baseline model is clear evidence that ICA artifacts selection can be easily automated using ML approaches. The novel aspects of the work include the algorithm for mu and alpha rhythm detection. The critical point is that the model is publicly available and additionally can be used as a pre-trained model for posterior modifications for other tasks.

Subjective labeling and ML training was performed on a dataset of ICs obtained on EEG data recorded in pre-school and school-age children, a population with usually many artifacts. This type of dataset is relatively unrepresented in the previous research on automatic IC extraction. The main work with infant ICA was done by Adjusted ADJUST algorithm (Leach et al., 2020) does not rely on machine learning techniques. The dataset consists of 630 ICA components acquired from 20 children, making up a unique publicly available dataset that can be used for various goals, e.g., for refitting new private models for ICA detection.

There are several points for future development of the project related to the annotation module and the ML module. The annotation module advances are related to the reorganization of available classes to mark into a hierarchical structure. Users can first select the artifact and specify it more precisely, for example, Eyes->Horizontal eye movements. Moreover, the first trial of expert annotations forces us to reestablish an expert policy and force them to choose no more than two IC classes to train our models using representative samples.

The ML module showed a high-performance rate for most classes. Although the Heart class was not detected, the reason for that is the lack of class representatives and a low agreement

between the annotators. Moreover, the Mu/Alpha rhythms and Eyes results were also obtained with fewer data samples. Nevertheless, the ALICE approach (including newly designed features for Mu and Alpha classes) showed good classification accuracy for ICs labeling even though the agreement between expert opinions was relatively poor. Still, for Heart and Channel Noise classes, none of the trained models worked well. Probably new robust predictive features or more complex ML models (i.e., based on convolutional neural networks) should also be developed to address these types of artifacts. We compared the performance of our algorithms with results reported in other studies. In Pion-Tonachini et al. (2019) authors report ROC curves with F1 scores. Eyes class F1 score is greater than 0.9, brain and muscles classes are in the range between 0.8 and 0.9, which is higher than results obtained using our model; at the same time, the heart class, like in our case, is reported as uninformative. In Tamburro et al. (2018), the authors reported accuracy, sensitivity, and false omission rates and provided complete data for eye movements, eye blinks, and muscle activity. The resulted F1 scores were greater than 0.9. In terms of our model, the low agreement between experts as an outcome of different labeling approaches might affect the final score.

Nevertheless, with additional datasets we discovered that the result can gain higher values for Eyes IC class with F1 score on the level of 0.87. Such values can be achieved for both adult EEG as well as for children EEG. This result implies that ALICE ML pipeline is robust to datasets of different ages. On the other hand, models require retraining to be suitable for data of different age or data of different ICA algorithm. This observation examined that database requires more components to show stable result over any type of dataset.

The current performance of ML algorithms in the ALICE toolbox is based mainly on two experts' estimations, whereas a manifold of professional annotations produces more objective estimates for components labeling. In future research, we aim to invite the wider expert community to label their datasets and expand current models' abilities or future models to define the functional nature of IC components. Thus, we encourage any reader to become a part of the ALICE project. More information about the potential contribution is provided on our web site http://alice.adase.org/docs/contribute.

To summarize, the main improvements implemented in ALICE as compared to previously developed toolboxes are the following:

- The ALICE toolbox allows not only detection of noisy IC but also automatic identifications of components related to the functional brain oscillations such as alpha and mu-rhythm.
- The ALICE project accumulates different benchmarks based on crowdsourced visual labeling of ICs collected from publicly available and in-house EEG recordings, resulting in a constantly growing high-quality IC dataset.
- ALICE implements the new strategy for consentient labeling of ICs by several experts.
- ALICE allows supervised ML model training and re-training on available data subsamples for better performance in specific tasks (i.e., movement artifact detection in healthy or autistic children).
- Finally, ALICE provides a platform for EEG artifact detection model comparison as well as a platform for neuroscientist self-assessment based on established performance metrics.

Thus, strength of the ALICE project implies the creation and constant updating of the IC database, which will be used for continuous improvement of ML algorithms for automatic labeling and extraction of non-brain signals from EEG. The developed toolbox will be available to the scientific community in an online service and open-source codes.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/supplementary material.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Ethics Committee of the Institute of Higher Nervous Activity and Neurophysiology or Russian Academy of Sciences. Written informed consent to participate in this study was provided by adult volunteers or the children participants' legal guardian/next of kin.

AUTHOR CONTRIBUTIONS

GS, AL, MN, OM, IP, GP, AR, OS, and MS conceived and designed the study. AL and GS developed the tool. AL, MN, and GS developed the model. OS, GS, and AR provided dataset. GP and AR analyzed the ICA data. OS, OM, MS, GS, AL, and MN wrote the manuscript. All authors revised the article.

FUNDING

Data collection, analysis and toolbox development was funded by the Russian Science Foundation (RSF), Grant No. #20-68-46042. Part of the work devoted to the application of machine learning models was made within funding from the Analytical center under the RF Government (subsidy agreement 000000D730321P5Q0002, Grant No. 70-2021-00145 02.11.2021). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

ACKNOWLEDGMENTS

The authors wish to thank Anastasia Neklyudova and Dina Mitiureva for the contribution to the initial stages of this research.

REFERENCES

Berger, H. (1931). Über das elektrenkephalogramm des menschen - dritte mitteilung. *Arch. Für Psychiatr. Und Nervenkrankheiten* 94, 16–60. doi: 10.1007/BF01835097

Chaumon, M., Bishop, D. V. M., and Busch, N. A. (2015). A practical guide to the selection of independent components of the electroencephalogram for artifact correction. *J. Neurosci. Methods* 250, 47–63. doi: 10.1016/j.jneumeth.2015.02.025

Chen, T., and Guestrin, C. (2016). XGBoost: a scalable tree boosting system. *arXiv* [preprint]. doi: 10.1145/2939672.2939785

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educ. Psychol. Measure.* 20, 37–46. doi: 10.1177/001316446002000104

Cuellar, M. E., and del Toro, C. M. (2017). Time-frequency analysis of mu rhythm activity during picture and video action naming tasks. *Brain Sci.* 7:114. doi: 10.3390/brainsci7090114

da Cruz, J. R., Chicherov, V., Herzog, M. H., and Figueiredo, P. (2018). An automatic pre-processing pipeline for EEG analysis (APP) based on robust statistics. *Clin. Neurophysiol.* 129, 1427–1437. doi: 10.1016/j.clinph.2018.04.600

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

Fleiss, J. L., and Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educ. Psychol. Measure.* 33, 613–619. doi: 10.1177/001316447303300309

Frølich, L., Andersen, T. S., and Mørup, M. (2015). Classification of independent components of EEG into multiple artifact classes. *Psychophysiology* 52, 32–45. doi: 10.1111/psyp.12290

Garakh, Z., Novototsky-Vlasov, V., Larionova, E., and Zaytseva, Y. (2020). Mu rhythm separation from the mix with alpha rhythm: principal component analyses and factor topography. *J. Neurosci. Methods* 346:108892. doi: 10.1016/j.jneumeth.2020.108892

Gramfort, A. (2013). MEG and EEG data analysis with MNE-python. *Front. Neurosci.* 7:267. doi: 10.3389/fnins.2013.00267

Kuhlman, W. N. (1978). Functional topography of the human mu rhythm. *Electroencephalogr. Clin. Neurophysiol.* 44, 83–93. doi: 10.1016/0013-4694(78)90107-4

Landis, J. R., and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics* 33:159. doi: 10.2307/2529310

Leach, S. C., Morales, S., Bowers, M. E., Buzzell, G. A., Debnath, R., Beall, D., et al. (2020). Adjusting ADJUST: optimizing the ADJUST algorithm for pediatric data using geodesic nets. *Psychophysiology* 57:13566. doi: 10.1111/psyp.13566

Lyakso, E., Frolova, O., and Matveev, Y. (2020). Speech features and electroencephalogram parameters in 4- to 11-year-old children. *Front. Behav. Neurosci.* 0:30. doi: 10.3389/FNBEH.2020.00030

Makeig, S., Bell, A. J., Jung, T.-P., and Sejnowski, T. J. (1996). *Independent Component Analysis of Electroencephalographic Data*. Cambridge, MA: MIT Press.

Marshall, P. J., Bar-Haim, Y., and Fox, N. A. (2002). Development of the EEG from 5 months to 4 years of age. *Clin. Neurophysiol.* 113, 1199–1208. doi: 10.1016/S1388-2457(02)00163-3

Mognon, A., Jovicich, J., Bruzzone, L., and Buiatti, M. (2011). ADJUST: an automatic EEG artifact detector based on the joint use of spatial and temporal features. *Psychophysiology* 48, 229–240. doi: 10.1111/j.1469-8986.2010.01061.x

Nolan, H., Whelan, R., and Reilly, R. B. (2010). FASTER: fully automated statistical thresholding for EEG artifact rejection. *J. Neurosci. Methods* 192, 152–162. doi: 10.1016/j.jneumeth.2010.07.015

Palmer, J., Kreutz-Delgado, K., and Makeig, S. (2011). *AMICA: An Adaptive Mixture of Independent Component Analyzers with Shared Components*. San Diego, CA: Technical Report, Swartz Center for Computational Neuroscience, 1–15.

Pedregosa, F., Gael, V., Gramfort, A., Michel, V., and Thirion, B. (2011). Scikit-learn: machine learning in python fabian. *J. Mach. Learn. Res.* 12, 2825–2830. doi: 10.1289/EHP4713

Pedroni, A., Bahreini, A., and Langer, N. (2019). Automagic: standardized preprocessing of big EEG data. *NeuroImage* 200, 460–473. doi: 10.1016/j.neuroimage.2019.06.046

Pion-Tonachini, L., Kreutz-Delgado, K., and Makeig, S. (2019). ICLabel: an automated electroencephalographic independent component classifier, dataset, and website. *NeuroImage* 198, 181–197. doi: 10.1016/j.neuroimage.2019.05.026

Robbins, K. A., Touryan, J., Mullen, T., Kothe, C., and Bigdely-Shamlo, N. (2020). How sensitive are EEG results to preprocessing methods: a benchmarking study. *IEEE Trans. Neural Syst. Rehabil. Eng.* 28, 1081–1090. doi: 10.1109/TNSRE.2020.2980223

Tamburro, G., Fiedler, P., Stone, D., Haueisen, J., and Comani, S. (2018). A new ICA-based fingerprint method for the automatic removal of physiological artifacts from EEG recordings. *PeerJ* 2018:e4380. doi: 10.7717/peerj.4380

Winkler, I., Haufe, S., and Tangermann, M. (2011). Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behav. Brain Funct.* 7:30. doi: 10.1186/1744-9081-7-30

Conflict of Interest: 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.

frontiers in Neuroinformatics

TECHNOLOGY REPORT
published: 07 January 2022
doi: 10.3389/fninf.2021.768615

The Neuroscience Experiments System (NES)–A Software Tool to Manage Experimental Data and Its Provenance

Margarita Ruiz-Olazar[1,2], *Evandro Santos Rocha*[1], *Claudia D. Vargas*[1,3] *and Kelly Rosa Braghetto*[1,4]*

[1] Research, Innovation and Dissemination Center for Neuromathematics, University of São Paulo, São Paulo, Brazil, [2] Polytechnic Faculty, National University of Asunción, Asunción, Paraguay, [3] Institute of Biophysics Carlos Chagas Filho, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, [4] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil

Computational tools can transform the manner by which neuroscientists perform their experiments. More than helping researchers to manage the complexity of experimental data, these tools can increase the value of experiments by enabling reproducibility and supporting the sharing and reuse of data. Despite the remarkable advances made in the Neuroinformatics field in recent years, there is still a lack of open-source computational tools to cope with the heterogeneity and volume of neuroscientific data and the related metadata that needs to be collected during an experiment and stored for posterior analysis. In this work, we present the Neuroscience Experiments System (NES), a free software to assist researchers in data collecting routines of clinical, electrophysiological, and behavioral experiments. NES enables researchers to efficiently perform the management of their experimental data in a secure and user-friendly environment, providing a unified repository for the experimental data of an entire research group. Furthermore, its modular software architecture is aligned with several initiatives of the neuroscience community and promotes standardized data formats for experiments and analysis reporting.

Keywords: neuroscience, experiment data, data management, data provenance, open-source software

OPEN ACCESS

Edited by:
William T. Katz, Janelia Research Campus, United States

Reviewed by:
Ryan Ly, Lawrence Berkeley National Laboratory, United States
Anna-Kristin Kaufmann, Swiss Federal Institute of Technology Lausanne, Switzerland

***Correspondence:**
Kelly Rosa Braghetto
kellyrb@ime.usp.br

Received: *31 August 2021*
Accepted: *13 December 2021*
Published: *07 January 2022*

Citation:
Ruiz-Olazar M, Rocha ES, Vargas CD and Braghetto KR (2022) The Neuroscience Experiments System (NES)–A Software Tool to Manage Experimental Data and Its Provenance. Front. Neuroinform. 15:768615. doi: 10.3389/fninf.2021.768615

1. INTRODUCTION

Although the overlap between neuroscience and informatics has been growing rapidly in the recent years, collection and organization of experimental data are still frequently done manually. A neuroscience experiment may involve the generation and manipulation of large amounts of both raw and processed data. There is a wide variability in the types of data that are collected, from the form and behavior of individual neurons to measures of brain functioning. This large quantity and variety of information requires a type of database that is especially designed for this purpose. However, the provenance information of raw data is too often lost or, when digitized, ends up as text files or spread-sheets without a standardized structure (Koslow, 2000). Within this context, the reproducibility of experiments—a core scientific principle—and the reuse of data may be seriously compromised. Efforts to develop best practices must be made on four foundational principles—Findability, Accessibility, Interoperability and Reusability (FAIR), as described by Wilkinson et al. in the FAIR Guiding Principles for scientific data management and stewardship (Abrams et al., 2021).

TABLE 1 | Software tools for neuroscience experiments management.

System	Focus	Ontology	Open Source
[a]EEGBase	EEG/ERP	OEN	Yes
[b]G-node	Neurophysiology	odML	Yes
[c]Psychopy	Experimental protocol	No	Yes
[d]Expyrement	Experimental protocol	No	Yes
[e]OpenSesame	Experimental protocol	No	No

[a]*https://eegdatabase.kiv.zcu.cz/*; [b]*http://www.g-node.org/*; [c]*http://www.psychopy.org/*; [d]*http://www.expyriment.org/*; [e]*http://osdoc.cogsci.nl/*.

This scenario calls for the use of computational tools to document each step of an experiment and to facilitate the electronic data capture. Although some tools have been developed and applied for this purpose (Mouček et al., 2014; Sobolev et al., 2014), there is still a lack of user-friendly software platforms that researchers can use to register different types of experiments and their working environment in a unified repository. These platforms should allow scientists to examine the data and metadata and know exactly how these were obtained, as well as how the experiment was performed. Such tools should be as easy to use as possible to reduce the time spent documenting experiments, while being able to support a wide variety of experimental designs (Peirce, 2009). Moreover, it should be platform-independent and a free/libre open-source software (FLOSS).

Addressing this problem, the Research, Innovation and Dissemination Center for Neuromathematics (NeuroMat), hosted by the University of São Paulo Research, Brazil[1] has developed the Neuroscience Experiments System (NES), a FLOSS that assists neuroscientists in the management of experimental data while providing provenance recording and interoperability. NES is a Web system that offers a user-friendly interface, allowing quick learning. Its data model combines several proposals from the scientific community for neuroscience data and metadata representation.

The NES modular structure provides functionalities for the registration of participant data and for experiment management. The participant registration functionality allows the collection and storage of personal and social-demographic data and medical evaluations. The experiment management includes experimental protocol registration (e.g., definition of tasks, stimuli, instructions, and configuration of equipment) and electrophysiological data and metadata storage. Presently, NES is equipped with modules allowing data collection from experiments performed in humans involving electroencephalography (EEG), electromyography (EMG), transcranial magnetic stimulation (TMS), and response times.

This article presents the approach used in the NES to manage data and metadata of neuroscience experiments. The NES innovative data model was designed to provide support for a wide range of experimental designs and to allow the efficient management of all steps of the experimental protocol and their different types of data.

The remainder of this article is organized as follows. Section 2.1 provides a brief characterization of the experimental data used in neuroscience, while section 2.2 analyzes some software tools for management of this kind of data. Section 2.3 describes the NES data model and the main functionalities it supports. Section 3 presents the NES software architecture and details about its implementation. It also presents an example of use of NES in the creation of an open database. Finally, the concluding remarks, including a discussion of the limitations of NES and future directions, are presented in section 4.

2. METHODS

2.1. Experimental Data in Neuroscience: Characterization

Designing an experiment includes a number of stages where the parameters and structure of the experiment are made clear. There are different types of neuroscience experiments (e.g., behavioral, cognitive, electrophysiological, and neuroimaging) with a great variability of experimental processes and a high heterogeneity of formats of collected data. An experimental process can be understood as comprising an experimental design and an experimental protocol. An experimental design includes the overall set-up of the experiment, in so far as it specifies the experimental context (e.g., how and where objects are to be arranged) and the materials and methods to be used (e.g., equipment settings). The experimental protocol is the set of step-by-step instructions that an investigator follows each time he or she runs an experiment (Sullivan, 2009). It includes a group of participants who will take part in the experiment. The steps of the experimental protocol can be performed sequentially or in parallel. After the approval of the experiment design and the experimental protocol, the group of participants is selected and the data collection starts.

The great heterogeneity of data collected in neuroscience experiments (e.g., EEG, EMG, fMRI, questionnaire responses) makes collaboration between members of the community difficult, since research groups would have to make a significant effort to standardize their lexicons and their data before collaboration could add value to such joint efforts (Hall et al., 2012). Furthermore, the information concerning the experimental process is too often lost or when digitized, it ends up becoming text files or spread-sheets without a standardized structure, or poor quality data with insufficient documentation. Sometimes, the data lacks metadata, standardized representation, or a legible structure (Barkhof, 2012).

To enhance the reproducibility of neuroscience studies, researchers need to know the precise acquisition parameters, the experimental conditions, and how the raw data were acquired. These different types of information, generally called *provenance information*, are metadata which is used to record the experimental process, the purpose of the experiment and details about its data results, as well as formal annotations and notes made by scientists.

[1]https://neuromat.numec.prp.usp.br/

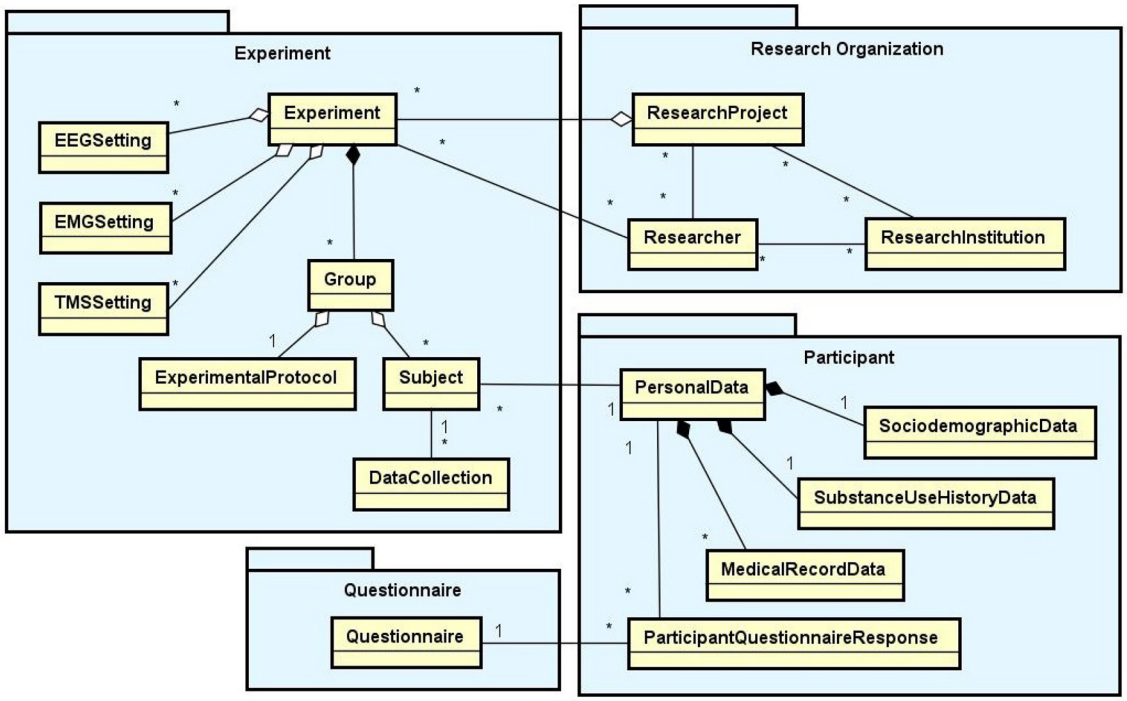


FIGURE 1 | Main data modules of NES (`Experiment`, `Research Organization`, `Participant`, and `Questionnaire`) with their entity types and relationships. In the diagram, the rectangles represent entity types, the lines ending with black diamonds represent composition relationships, the lines ending with white diamonds represent aggregation relationships, and the conventional lines represent association relationships. The "1" and "*" (that means "many") near the lines indicate the cardinalities of the relationships. For example: a `Group` aggregates one `ExperimentalProtocol` and many `Subjects`; an `Experiment` is composed of one or more `Groups`; a `Subject` can have many `DataCollections`, but a `DataCollection` is associated with only one `Subject`.

A unified data model for handling metadata is still an open research problem. The problem is compounded when the volume of collected data begins to grow. Unlike the progress in workflow-based systems, which provide consistent mechanisms to manage the provenance of derived data generated through scientific workflows, the availability of open data models and free software tools to support raw data routine collection is limited. Thus, the creation of standardized models and formats for representing and storing raw data and its provenance information is not a trivial task and depends on collaborative efforts from the neuroscience community (Ruiz-Olazar et al., 2016).

Due to the great variability in experimental processes and the heterogeneity of collected data formats, neuroscience experimental raw data and information relating to provenance require specific and innovative ways of representation and storage. The guidelines of Gibson et al. (2008), Poldrack et al. (2008), and Frishkoff et al. (2011) include information that is considered important for data analysis and for understanding the experiment performed. However, these guidelines are neither complete data representation models nor data storage models.

2.2. Related Software Tools

A brief review of the open-source software tools created to support the management of neuroscience experiments shows that most of these systems can be divided in two groups: (i) those that focus on the storing and sharing of electrophysiological data and (ii) those that focus on the management of experimental protocols. Some of those that are in the first group provide interfaces to manipulate electrophysiological data objects, such as data arrays, events, regions of interest, etc., or extensively annotate these specific data objects. Those in the second group provide the management of the experimental protocol, accurate presentation of stimuli, and mechanisms for the collection of participant responses. Software tools from the two groups can be combined in order to help researchers in their data collecting routine throughout a neuroscience experiment. **Table 1** compares some software tools for neuroscience experiment management.

Among the software tools that are in the first group is EEGBase (Mouček et al., 2014), which was designed to enable data exchange based on files. The EEGbase is a system for storage and management of EEG/ERP (electroencephalography, event-related potentials) resources, such as data, metadata, analysis tools, and documents related to experiments in respect of EEG/ERP. It provides the possibility to work offline by using a client-server approach, and data and metadata can be registered using a tablet or mobile phone based on a client-mobile system. These platforms can synchronize data with the EEGBase Web-based portal. Through this portal, researchers can store, manage, search, and share EEG/ERP

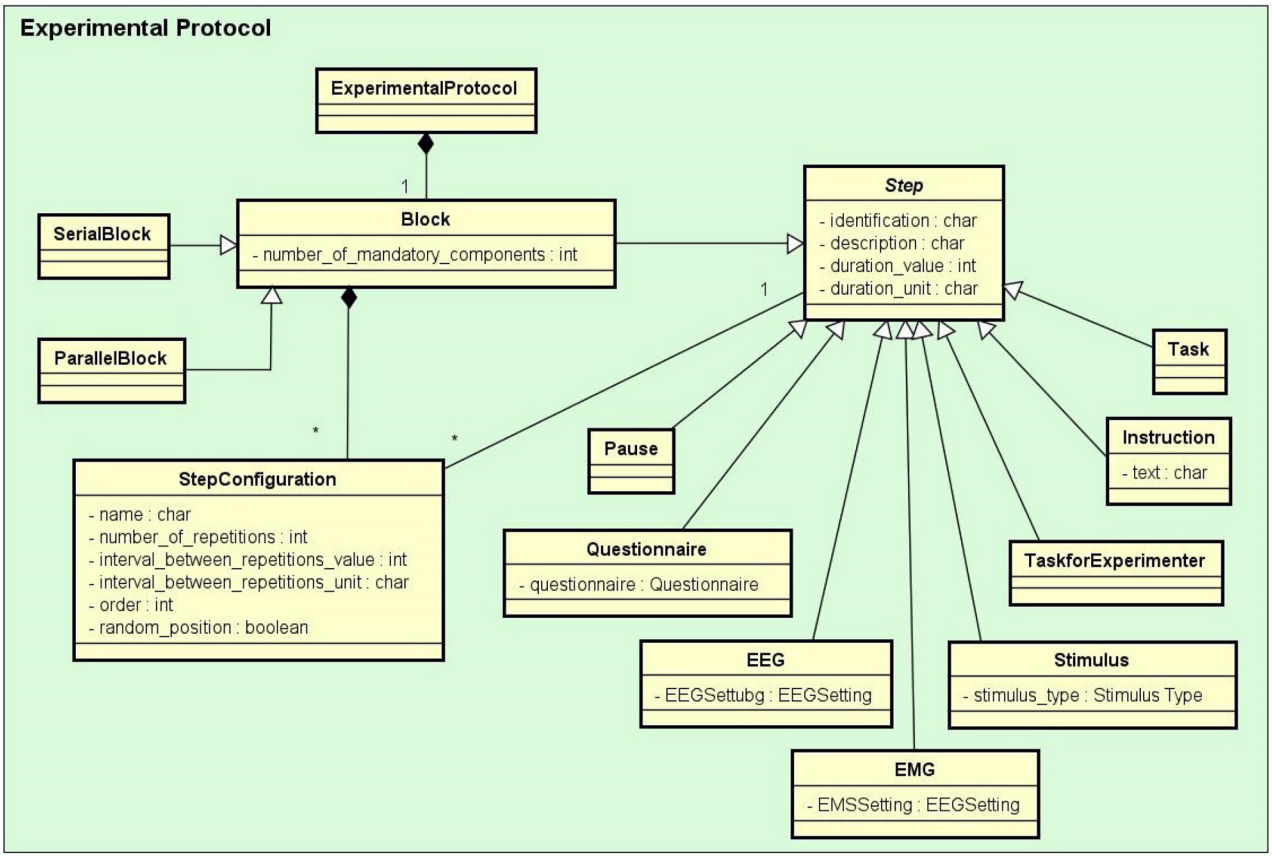


FIGURE 2 | The Experimental Protocol conceptual data schema. In the diagram, the rectangles represent entity types and some of their main attributes, the lines ending with black diamonds represent composition relationships, and the lines ending with triangles represent inheritance relationships. For example: `Stimulus`, `Task`, `Questionnaire`, and `EEG` are subtypes of `Step`, each one of them inherits the properties of `Step`. An `ExperimentalProtocol` is composed of a `Block` (of blocks) of `StepConfigurations` which define how the protocol `Steps` are performed.

data. The data and metadata are implemented according to a defined ontology and registered using predefined HTML forms. However, the metadata is registered in textual mode.

The German Neuroinformatics Node, G-node (Sobolev et al., 2014), provides a data management system with interfaces to operate with electrophysiology raw data objects. G-node is a data platform and Python library that implements tools, standards, and conventions established in an electrophysiological context. This approach is based on combining a standardized data model, NEO (Garcia et al., 2014), with a flexible and extensible metadata format, odML (Grewe et al., 2011). OdML uses the open metadata Markup Language to annotate data with information about the stimulus, data acquisition, and experimental conditions. In contrast, its extensible "key-value pairs" format does not specify the relevant information that should be registered, but it depends on the experimenter. NEO provides a flexible method of manipulating neurophysiological data and its I/O library can read a wide range of neurophysiological file formats. However, it cannot currently read relevant information such as the number of used channels, sample rate, frequency, etc. In addition, G-node offers integration with other Python tools that use these data models. However, these data models focus on cellular and intra-cellular experiments, without providing a comprehensible data schema to represent electrophysiological data such as EEG and EMG results.

Among the software tools that allow management of the experimental protocol, accurate presentation of stimuli, and collection of participant responses are Psychopy (Peirce, 2007) and Expyrement (Krause and Lindemann, 2014). Both provide an open-source software library that allows a very range of visual and auditory stimuli and a great variety of experimental designs to be generated within a framework based on Python. Expyrement aims at designing and conducting behavioral and neuroimaging experiments. Nevertheless, they do not offer a graphical interface that many users have come to expect. These packages require some effort from the users in respect of writing scripts in standard Python syntax to generate a variety of behavioral experiments.

Another related tool is OpenSesame (Mathôt et al., 2012), which provides a graphical and scripting interface to create a wide range of experiments, including psychophysical

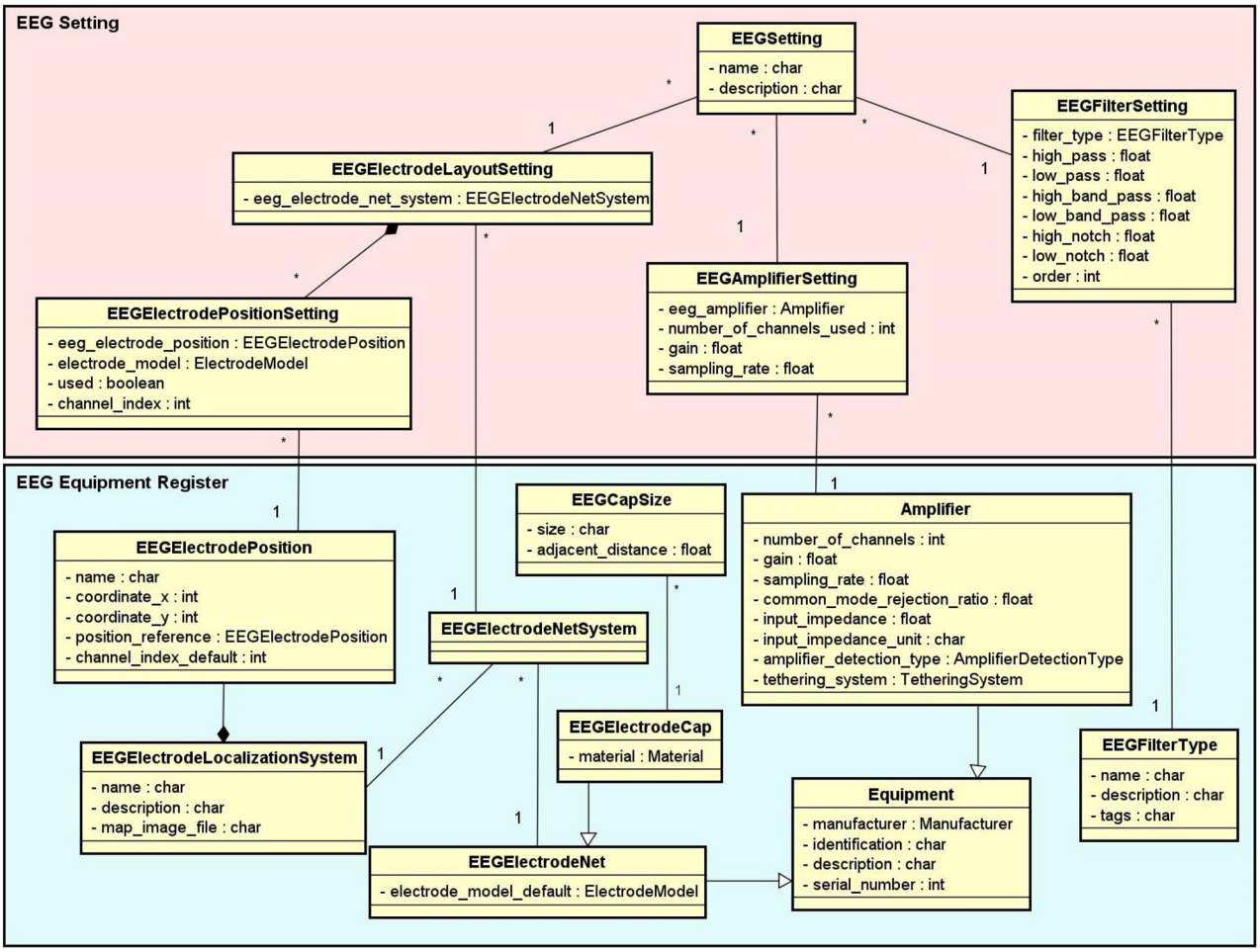


FIGURE 3 | EEG Setting and Equipment Register conceptual data schemas. The EEG Equipment Register schema contains the entity types with the technical characterization associated to a given EEG setup, such as amplifier, filter, electrode net, electrode cap, electrode localization system, etc. The EEG Setting schema contains the entity types which represent the configurations of the EEG equipment used in each step of an EEG experiment.

experiments, speed response time tasks, eye-tracking studies, and questionnaires. Some kinds of tasks need to be defined using Python scripting, since the tool does not provide a good graphical interface to support their definition.

The software packages described above focus on specific types of scenarios and fail to describe other types of experimental protocols. Although they provide models to store data and metadata, these models are very extensible, making the subsequent generation of queries to track the provenance information in the experiment more difficult. This information is frequently written in a non-understandable form, hindering its interpretation by other experimenters who cannot therefore later reproduce or verify the findings (Ruiz-Olazar et al., 2016). The users of software packages that are part of the second group require technical knowledge to write scripts in Python to define the experimental protocols and later execute them. Neuroscience labs need tools with as wide a range of experimental designs as possible that assist the experimenter in the management of all steps of the neuroscience experiments, without being required to have knowledge of a variety of software and programming languages to be able to use them.

2.3. The NES Data Model and Main Functionalities

Based on an exhaustive literature research and interviews with domain specialists, we have identified the requirements a software tool should satisfy to support data management in the daily lab routine. In this section, we present the set of functionalities implemented in NES to meet these requirements and the data model that support them. These functionalities provide a complete interface for the storage and management of data and metadata from all the steps of a neuroscience experimental protocol. They are related to a set of database modules represented in the diagram in **Figure 1**: `Experiment`, `Research Organization`, `Participant`, and `Questionnaire`. The diagram shows

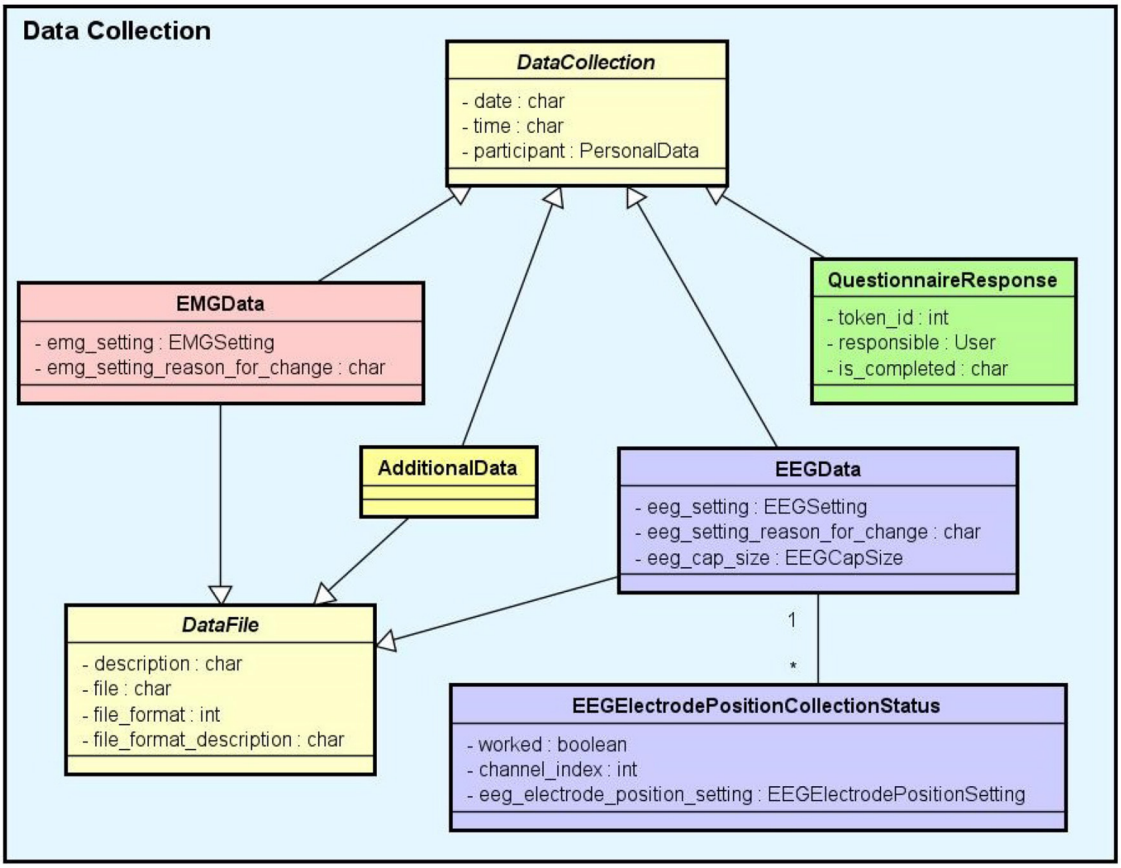


FIGURE 4 | Data collection conceptual data schema. There are several subtypes of data collection: EEG data, EMG data, questionnaire responses, and additional data (for other unlisted data types). Except for the questionnaire responses, all the data collections are files uploaded to the system.

the main entity types of each module and the relationships between them.

According to the NES data model, a research project can have one or more experiments. Each experiment is composed of equipment configurations and one or more groups of subjects, i.e., the individuals that take part in the experiment. Each group can have its own experimental protocol, which is composed of a set of steps. Moreover, each item of data collected in an experiment is associated with a specific step executed in the experimental protocol and the subjects who took part in it. For this reason, to be able to start storing the primary data collected in an experiment in NES, the researcher first needs to register in detail each step involved in the experimental protocol (e.g., the specific preparation for the realization of the experiment).

The NES database modules were designed to store the kinds of data whose structure is common to all experiments, i.e., data that can be described in terms of a standardized structure defined by a database schema. The data model used in NES is aligned with several formats used in neuroscience, enabling interoperability with the most promising initiatives for standardization of data representation for electrophysiology, as much as with guidelines to report neuroscience experiments (e.g., MINI Gibson et al., 2008, MINEMO Frishkoff et al., 2011, and fMRI Poldrack et al., 2008). NES is able to manage several types of electrophysiological data and metadata used by the neuroscience community.

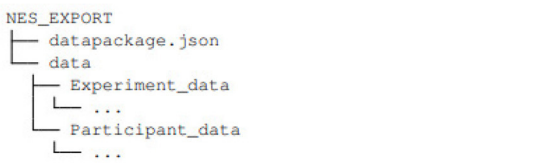
In the following sections, we provide more details about the database schema of each module. **Figures 1–4** are conceptual database schemas expressed using UML Class Diagrams. They purposely abstract details about the real database structure in order to make the diagrams easier to read and understand.

2.3.1. Participant Module

The `Participant` module supports functionalities to manage information related to the participants in the experiments. Its data schema specifies attributes of the participants that are significant for the experiments' design and interpretation.

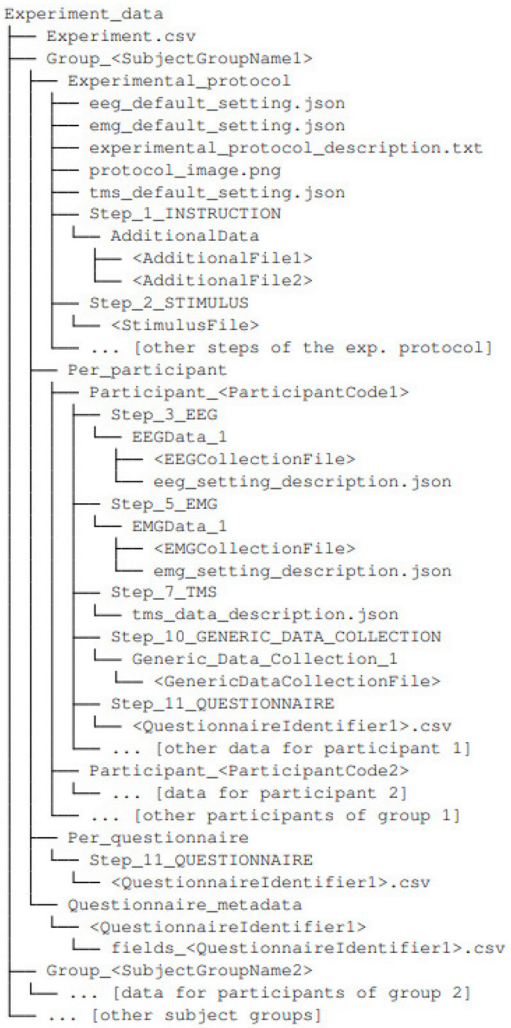
In the `Participant` data schema, as can be seen in **Figure 1**, the participant data is divided in five components:

- *Personal data* contains participants' basic information—identification, name, gender, birth date, address, and phone numbers.
- *Social Demographic data* registers the participants' native country, occupation, religion, and race.



A Top of the directory hierarchy of an export in NES.

B Structure of the `Participant_data` directory.



C Structure of the `Experiment_data` directory.

FIGURE 5 | Directory structure of an experiment dataset exported from NES (adapted from Peschanski et al., 2020). The root directory **(A)**, `NEX_EXPORT`, contains the directory `data`, with the data package files, and the file `datapackage.json`, which describes the structure and contents of the data package (as prescribed by the Frictionless Data standard). The experiment data is organized in two directories: `Participant_data` **(B)** and `Experiment_data` **(C)**. The former contains the data from the participants (including their responses for questionnaires applied outside the context of experiments), while the latter contains all data collected within the steps of the experimental protocol (and their metadata).

- *Substance Use History data* allows to register participants' history of use of alcohol, tobacco, and other drugs.
- *Medical Evaluation* allows storage of participants' medical records (including diagnosis with ICD codes and medical tests).
- *Questionnaire Response* includes the participants' answers to the questionnaires that are part of experimental protocols.

2.3.2. Experiment Module

The `Experiment` module is the core of NES. It supports functionalities for experiment registration and configuration as well as data collection. Its data schema was designed to be able to represent the structure and data of types of experiments frequently performed in humans, in addition to data stored in other widely used formats in neuroscience without any loss of information. For this purpose, the data schema organizes data related to the experiments, the groups of subjects, the steps of their experimental protocols, and the equipment settings. Each of these types of entities is described in more detail below.

- *Experiment*: It stores information that identifies the experiments and their purposes, the responsible researchers, and the resulting publications.
- *Experimental Protocol*: An experimental protocol is modeled in NES as a workflow composed of blocks of parallel or

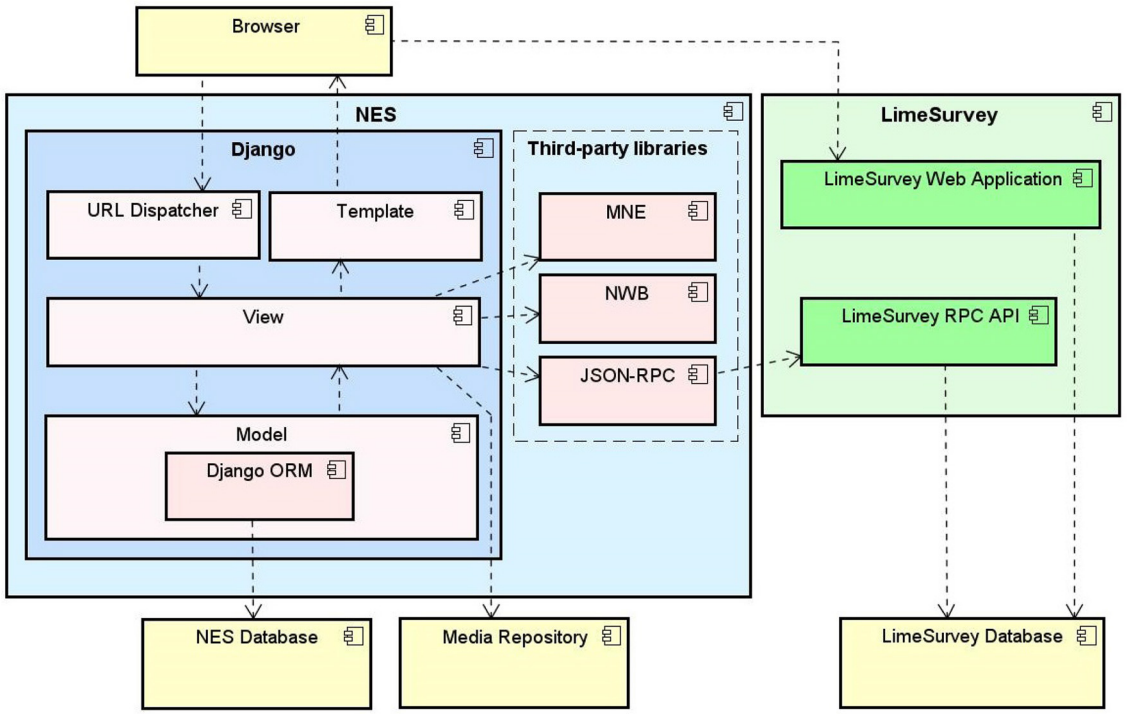


FIGURE 6 | NES software architecture, composed by the Django Web framework components and third-party libraries to convert data and provide interoperability with other tools.

sequential steps. There are several types of steps that a block may contain (e.g., task, stimuli, TMS, instructions, EEG, EMG, questionnaire administration, and other types of data collection). Each block or step can have its own configuration, such as the number of times it must be repeated, the time interval between repetitions, and its order in the protocol workflow, which can be deterministic or random. **Figure 2** shows the conceptual schema of the experimental protocol data.

- *Group*: Several groups of subjects can be created for an experiment. Each group is associated with an experimental protocol and the participants who take part of it, as shown in **Figure 1**.
- *Equipment Setting*: It stores information about equipment and materials of an eletrophysiology experiment. Each setting is associated with a step in an experimental protocol. NES can store the settings of the equipment used to record raw data, as well as the type of materials used in each data acquisition procedure. For EEG settings, as shown in **Figure 3**, NES can store amplifier settings and filter settings. The amplifier settings include gain, number of channels, common mode rejection rate, input impedance, and unit of impedance, among other information. The filter settings include information about the filter type, high pass cutoff, low pass cutoff, and order.

  Another important item of information considered in the Equipment Setting data schema is the electrode layout. NES allows electrode settings to be recorded individually or using an electrode net system, as the 10–20 system[2]. It also allows the registration of the electrode model, the electrode positions and their channel index. The spatial coordinates of each electrode, its position reference, and its default channel index can also be registered.

  For EMG experiments, in addition to storing information about the equipment and materials, NES can store systems for electrode placement, such as the SENIAM system (Surface ElectroMyoGraphy for Non-invasive Assessment of Muscles). Additionally, a list of muscles and its subdivision where the electrodes will be located can be recorded.

  For the Transcranial Magnetic Stimulation (TMS) setting, NES can store data about the TMS device and the coil model used in the experiment.
- *Data Collection*: This is another key component of the `Experiment` module. It supports the functionalities related to the management of data collected during the execution of the experimental protocol steps. NES attaches the data collection for each step of an experiment to the subject who took part in it. As shown in the conceptual schema of **Figure 4**, NES is able to handle several types of data collections: raw data obtained from a signal acquisition equipment (e.g., EEG and EMG), questionnaire responses, and any other type

[2]The 10–20 system is an internationally recognized method to describe and apply the location of scalp electrodes in EEG experiments.

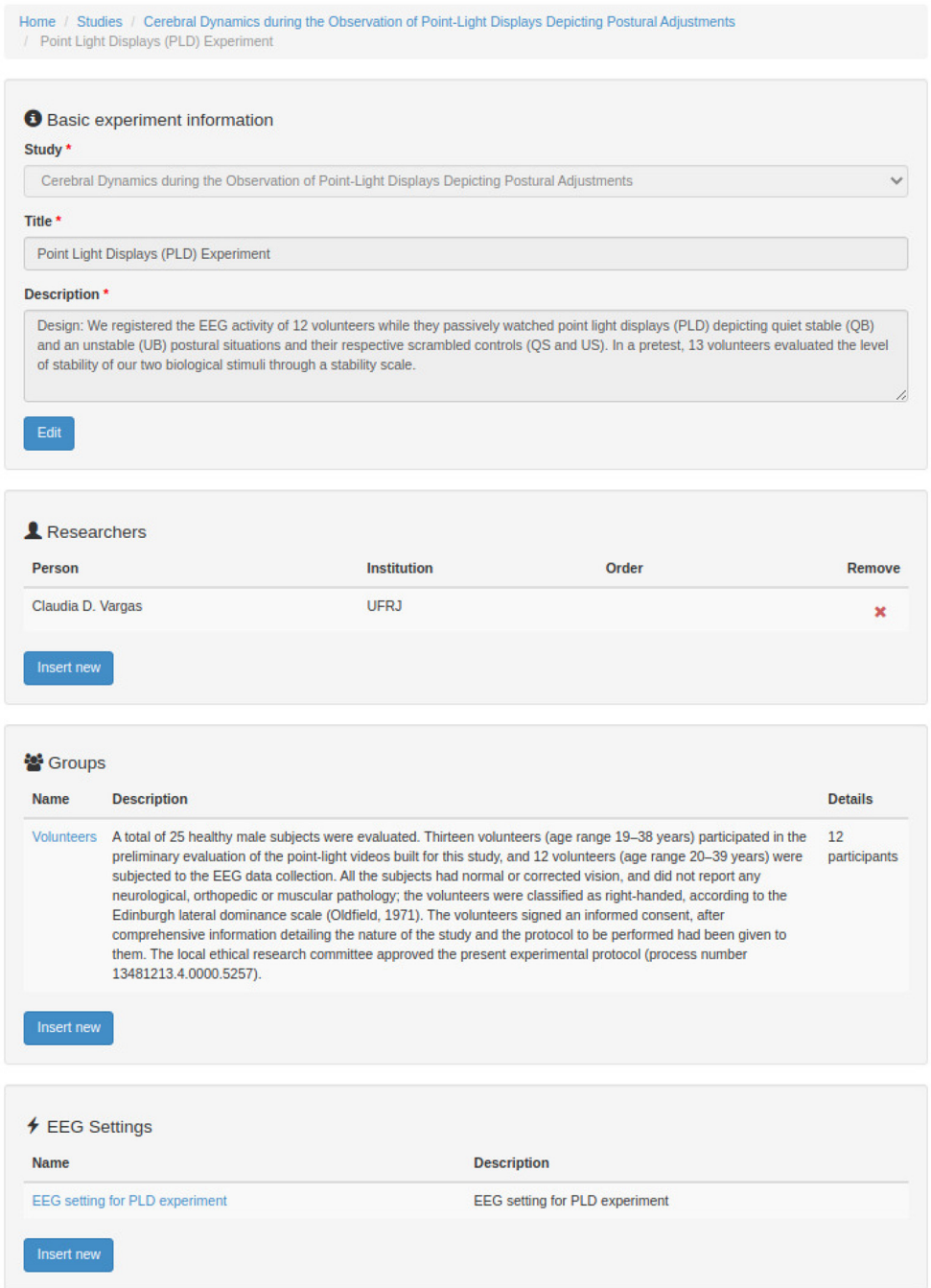


FIGURE 7 | NES graphical interface for registering the basic information about an experiment.

of additional files (such as spreadsheets, videos, and textual notes) that can be collected or generated in an experiment. The possibility of uploading additional files is also useful to store processed data alongside the raw data collected during the experiment. For example, a researcher can add one or more data collection steps at the end of the protocol of an experiment specifically to register the data (files) derived from the raw data collected in the previous steps. Information about how the derived data was produced can be registered in the form of textual notes.

The data model allows information about the file type format, and the date and time of the acquisition to be stored. In the case of EEG data acquisition, it is possible to record, for each participant, the size of the electrode cap and information

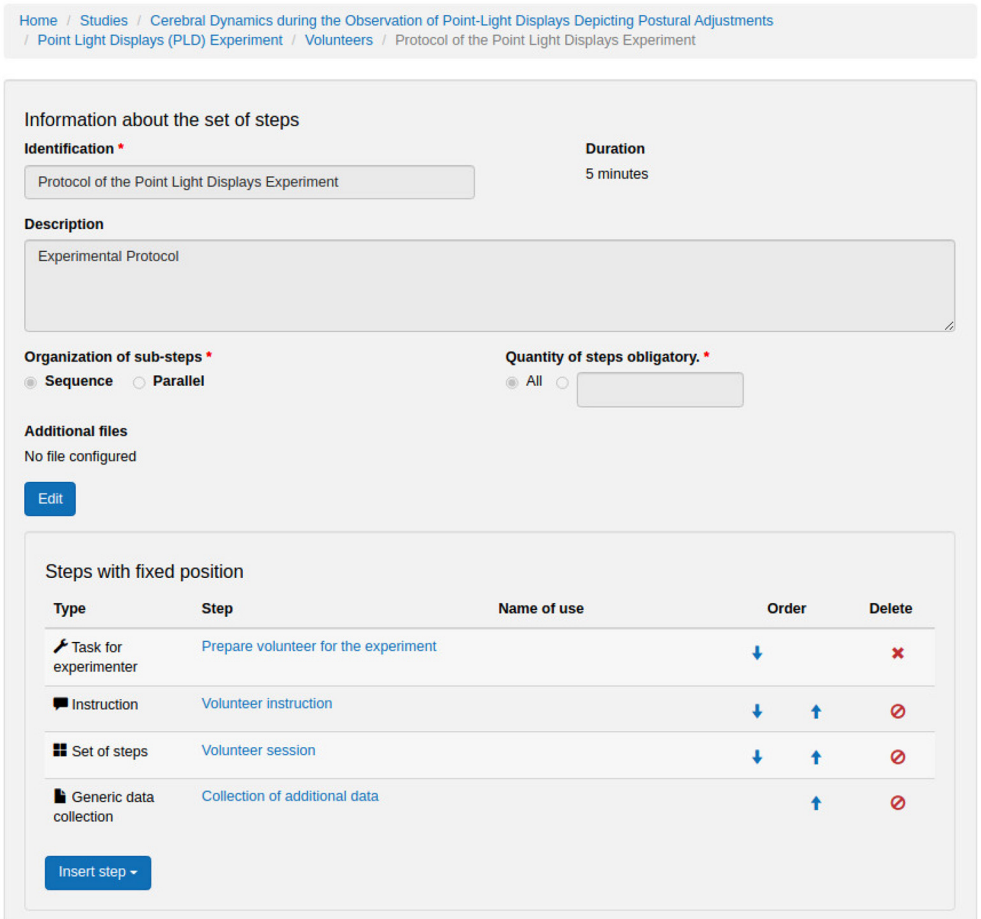


FIGURE 8 | NES graphical interface for registering an experimental protocol for a group of participants.

related to electrode positions and their status (e.g., used or not used) at the moment of the capture. The settings of the equipment and materials used in each data acquisition procedure can also be stored. This information is fundamental to enable the sharing and reuse of the raw data.

In NES, it is possible to create copies of an experiment (with or without the associated data collections). The copies are fully accessible through the user interface. This is useful for versioning control. One can also export an experiment and generate a .zip file with all the experiments' data and metadata. The .zip file can be later imported into other studies or NES installations. Other details about the export features are provided in section 2.3.5.

2.3.3. Questionnaire Module

Questionnaires are a very flexible way to collect data from study participants. In NES, a questionnaire can be configured as a step of an experimental protocol.

As questionnaires vary from study to study, they are difficult to be stored in a rigid, fixed database structure. To conveniently deal with this problem and also to provide more quality and security to data collected through questionnaires, a questionnaire management system, the LimeSurvey[3], was integrated into NES. This kind of system is a powerful, easy-to-use tool to create electronic question-and-answer surveys.

2.3.4. Research Organization Module

The `Research Organization` module supports functionalities to register information of the researchers that are working on the studies, their projects and the institutions involved. Its data schema stores data about the researchers, laboratories, and projects associated with the experiments. NES allows multiple experiments within the same research project, where each experiment can involve a different group of researchers. Researchers participating in an experiment have their own access and permissions to manage the experiment in NES.

NES implements a role-based control access (RBAC) approach to restrict system access to authorized users. RBAC is defined around roles (of users) and permissions (to perform particular system functions). Multiple permissions can be granted to a role, and a user can have multiple roles. Thus, it is simple

[3]https://www.limesurvey.org/

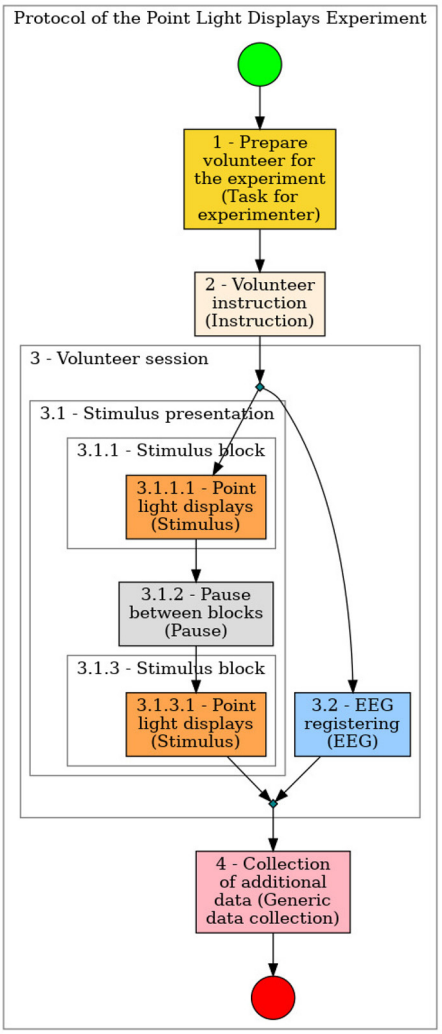


FIGURE 9 | Example of diagram generated by NES from an experimental protocol definition.

to manage permissions since users can be grouped according to their roles. NES has some default groups of users (such as Administrator, Junior Researcher, and Senior Researcher). However, using the Administration Interface, one may add or change roles and the permissions for each role to best suit the needs of the research institution. Examples of permissions which can be attributed to groups are: "Can view research project," "Can add study," "Can change subject," "Can delete survey," and "Can export experiment."

2.3.5. Data Export Functionality

An important functionality provided by NES which is connected with all its data modules is the export of experimental data and metadata. This functionality includes the data from participants (e.g., clinical diagnoses, socio-demographics), the data collected in the experiment execution (e.g., questionnaire responses, electrophysiological raw data), and metadata about the experimental protocol (e.g., description of the purpose of the experiment, description of the protocol steps, equipment configuration, and notes made by researchers).

Through the NES export functionality, a researcher is able to download experimental data and metadata in interoperable formats. It was implemented within the Frictionless Data philosophical and technical framework in order to decrease friction that is commonly associated with understanding data and metadata (Peschanski et al., 2020). Frictionless Data is an open-source framework for building data infrastructure. It was established by the Open Knowledge Foundation[4] to provide technical support to open science strategies. The framework includes various data standards to help to describe data. Its core specification, the Data Package, is a container format used for storing metadata alongside a dataset expressed as a simple JSON file named `datapackage.json` (Fowler et al., 2017).

NES offers two types of file organization in an export: per participant and per experiment. In both types, one can filter the group of participants whose data will be exported. The participants can be filtered, for example, by gender, location, diagnosis, and age. One can also select the participant data fields to be included in the export. For experiments with questionnaires, it is possible to choose the questions to be included.

Figure 5 shows the directory structure of an experiment dataset exported from NES. The structured data is exported in plain-text files in the CSV (*comma separated values*), containing both textual and numeric data. The equipment configuration is exported in JSON format. The EEG raw data can also be exported in the Neurodata Without Border (NWB 1) format (Teeters et al., 2015), a prominent initiative for standardization of data representation for neurophysiology. An NWB 1 file consists of several main groups, each of which is a container (similar to a directory) for different subsets of the data. In NES, the data included in the NWB 1 file is organized in three main groups: *General metadata*, *Device configuration*, and *Data acquisition*. *General metadata* contains information about the description of the experiment and some demographics data of the participants [e.g., experiment, subject id, sex, genotype (flesh tone), subject (natural of)]. *Device configuration* group contains the settings of the devices used in the EEG data collection (e.g., amplifier, filter device, electrode net layout). The *Data acquisition* group contains the raw data and metadata collected for each session of EEG (e.g., data, time, number of samples, electrode indexes, number of channels).

3. RESULTS

3.1. The Developed Software System

The functionalities and the data model described in section 2.3 were implemented in a platform-independent Web system, whose architecture is depicted at a high level in **Figure 6**. It is a standard three-tier Web architecture, with a data storage tier, an application layer, and a presentation tier.

[4]https://okfn.org/

The data storage tier is implemented as a relational database in the open-source database management system PostgreSQL[5]. The database is used to store all the structured data, according to the models described in section 2.3. Physical files, such as the EEG and EMG recordings and user additional data, are stored in the file-server system.

The application layer consists of a group of libraries that perform the execution engine of the system in the server side. NES was implemented in Python, using the Django Web framework[6]. This platform offers many benefits, such as scientific libraries, extensive documentation, and an active community. Python is an open-source programming language that has become one of the most popular programming language used in neuroscience systems.

NES uses third party libraries (e.g., MNE[7], NWB[8] and JSON-RPC[9]) to provide interoperability with other neuroscience tools. The MNE is an open-source Python software for exploring, visualizing, and analyzing human neurophysiological data such as Magnetoencephalography (MEG), EEG, sEEG and more. NES uses MNE to read and visualize several raw EEG data formats. The NWB-API is a Python API that NES uses to create NWB 1 files.

NES integrates with LimeSurvey to support the use of electronic questionnaires to collect data in experiments. LimeSurvey is an open-source, Web server-based software. It enables the storage of all data collected through the questionnaires in a "private" server. It relies on an underlying database management software which can be deployed on a server that is deemed appropriate to store the target data and customized to support different data access policies. With this structure, NES has full control over LimeSurvey data storage and access. NES communicates with the LimeSurvey application through the RemoteControl 2 API[10]. This API is a XML-RPC/JSON-RPC based Web service which offers several functions for questionnaire management.

Users access NES via a browser. The presentation tier is implemented using the Twitter Bootstrap[11] framework to generate the application layout and make it responsive, adjusting the Web pages dynamically according to the device used (e.g., desktop, mobile, tablet). Additionally, JavaScript was used to facilitate the implementation of some functionalities.

NES is an internationalized software, it can be adapted to various languages and regions without engineering changes. Currently, NES is localized to Brazilian Portuguese (pt-br) and English (en), but it can be localized to other languages by simply translating text and adding locale-specific components.

Figures 7–12 show some screenshots from the NES Web interface. **Figure 7** presents the NES page for registering the basic information about an experiment. It enables users to include groups of subjects for the experiment and equipment settings for the electrophysiology data captures. **Figure 8** shows the page for the definition of the experimental protocol of a group of subjects of an experiment. The experimental protocol is described as a workflow, which can contain blocks of sequential or parallel steps of several types. An example of workflow diagram NES automatically generates from a protocol definition is shown in **Figure 9**. In the protocol illustrated by the diagram, the *Stimulus presentation* and the *EEG registering* are, respectively, a block of stimulus steps and a data collection step which are performed in parallel in the experiment, according to the protocol. **Figure 10** shows the page for listing participants and data collections of an experiment. In **Figure 11**, the NES interface shows the position of the electrodes used in an EEG data collection. The interface allows the working electrodes used in the EEG recording to be indicated. Finally, **Figure 12** shows some configurations a user can set in the export of experimental data and metadata.

NES is licensed under the Mozilla Public License version 2.0 and its source code and documentation are available at https://github.com/neuromat/nes. The online software documentation and User Guide, with comprehensive descriptions of the NES functionalities, are available at https://nes.readthedocs.io/en/latest/.

3.2. Using NES to Create Open Databases

NES is a software that can be installed by a laboratory or a research group to locally manage experiments and their data. Besides, NES can also be used to support the generation of well-documented, anonymized datasets that can be published to openly share experimental data. An example of such an application can be seen in the NeuroMat Open Database (NeuroMat DB)[12], an initiative that provides an open-access platform for sharing and searching data and metadata from neuroscience experiments. Electroencephalographic (Martins et al., 2017; Hernández et al., 2021) and clinical data (Patroclo et al., 2019; Ramalho et al., 2019) collected within NeuroMat and organized in NES were made publicly available through the NeuroMat DB Web portal. **Figure 13** shows a page that lists these datasets.

Through NES, a researcher is able to send data and metadata of his/her experiments to the Open Database (as illustrated in **Figure 14**). Before sending the data, NES replaces the experiment participant's identifiers with random numbers in order to anonymize them. Personally identifiable information such as name, document number, address, and phone number can not be sent. In principle, the researcher is responsible for choosing among the study data what is going to be sent to the portal. All data is cryptographed upon transmission to the Open Database. When a new dataset arrives at the Open Database, it is evaluated by a curatorial committee who decides whether it is appropriate for publication. The committee then guarantees that no sensitive information will be made publicly available in the open database. If approved by the committee, the dataset is published on the portal.

[5]https://www.postgresql.org/
[6]https://www.djangoproject.com/
[7]https://mne.tools/
[8]https://pynwb.readthedocs.io/
[9]https://www.jsonrpc.org/
[10]https://manual.limesurvey.org/RemoteControl_2_API
[11]https://getbootstrap.com/
[12]http://neuromatdb.numec.prp.usp.br/

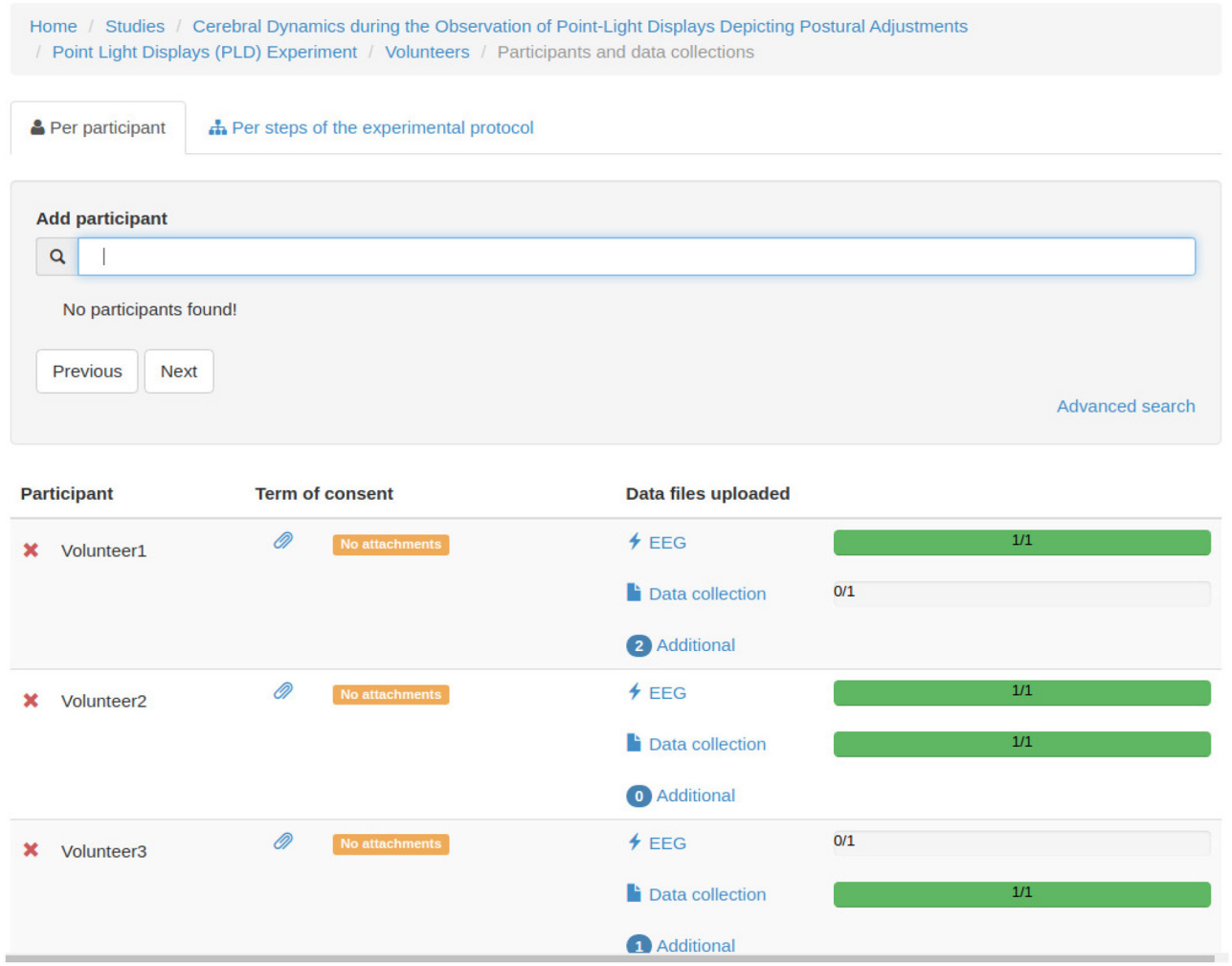


FIGURE 10 | NES graphical interface for listing participants and data collections of an experiment.

3.2.1. The Brachial Plexus Injury Database: A Case Study

The Center for Research in Neuroscience and Rehabilitation (NPNR) of the Deolindo Couto Institute of Neurology (INDC) at the Federal University of Rio de Janeiro (UFRJ), in collaboration with NeuroMat, investigated the brain plasticity that follows traumatic brachial plexus injury (TBPI) and its surgical reconstruction. The brachial plexus is composed of a set of peripheral nerves responsible for the sensory, motor, and autonomic innervation of the upper limbs. Injury to peripheral nerve structures and/or medullary avulsion as a result of a TBPI lead to changes in cortical representations and are also often associated with neuropathic pain (Torres et al., 2019). In recent years, the frequency of this type of injury (mainly caused by motorcycles accidents) has grown considerably in developing countries and has already become a public health concern. NPNR is using NES to collect, store and manage data from the TBPI studies, which are mainly made up of electrophysiological recording, responses to clinical questionnaires, and behavioral data from more than 170 patients. An anonymized portion from the TBPI database on NES was made publicly available on the NeuroMat Open Database Web portal[13]. As far as we know, this is the first worldwide open digital database centered on adult TBPI (Patroclo et al., 2019).

4. DISCUSSION

We identified the guidelines and models most widely used by neuroscientists in the representation and storage of experimental data (Ruiz-Olazar et al., 2016) and incorporated them in NES. To the best of our knowledge, there are no other open-source software tools which provide facilities to record the data and metadata involved in all steps of a neuroscience experiment.

[13]https://neuromatdb.numec.prp.usp.br/experiments/brachial-plexus-injury-database/

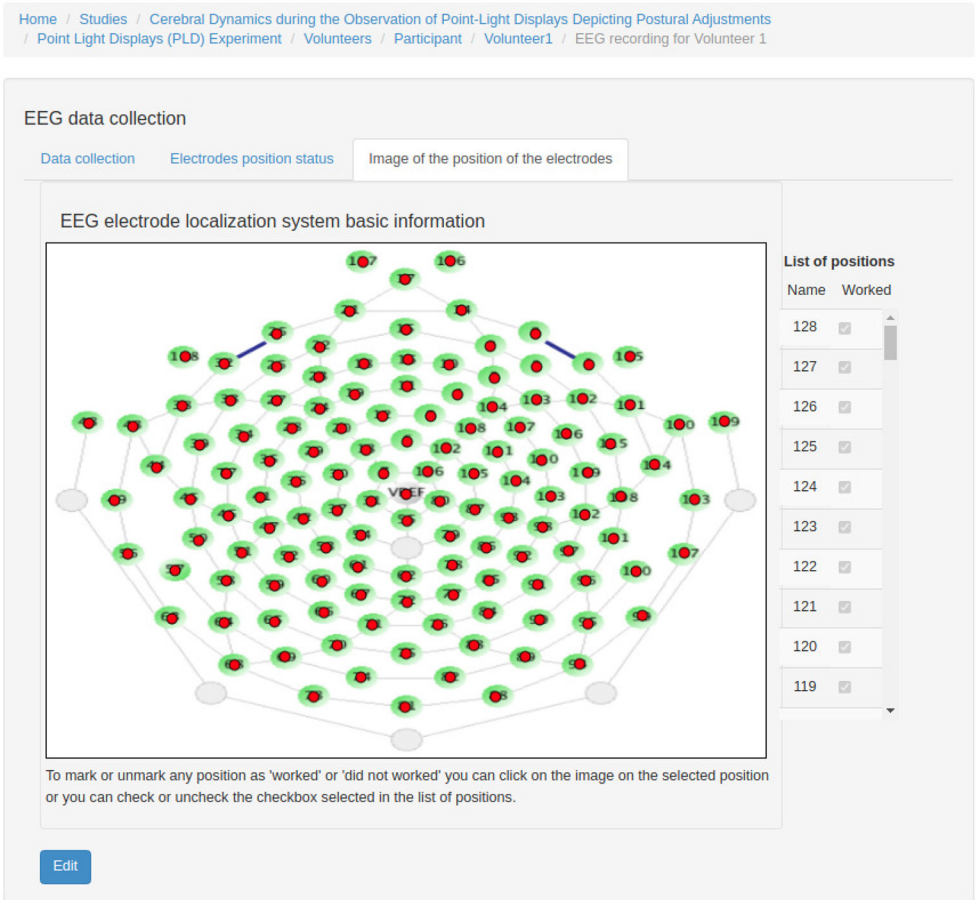


FIGURE 11 | NES graphical interface for registering electrode status in EEG data collection.

NES provides a structured and comprehensive platform with robust tracking of data provenance that is fundamental to enable the reproduction of the experiment. It was developed to keep together experimental data and information on its provenance, defined by the seven W's (Who, What, Where, Why, When, Which, (W) how). Examples of provenance information maintained by NES are: information about the scientists responsible for the experiment and collection of data and the description of the subject groups (who); the details about the recording protocol or behavioral data collection (e.g., the types of data collection performed) (what); the details of the experimental protocol used in the collection of primary data (how); the start/end date-time for data collection (when); the purpose of the experiment (why); the information about the experimental conditions to which the groups of subjects are submitted, such as tasks performed and stimuli applied (which); the information about the laboratory where data was collected (where) and even publications or other results that have arisen from the study of the collected data. Scientists can also record additional details for each participant in the experiment, such as information about his/her clinical history and social-demographic data.

It is worth mentioning that NES is not a new way to standardize the representation of experimental data. There are several models and formats (e.g., NeoHDF5 Garcia et al., 2014, NWB Teeters et al., 2015, and NIX Stoewer et al., 2014) currently in development to address this issue. These models are appropriate for organizing and exchanging data of a particular type and from a particular experiment. However, they do not replace the function of a database system, as provided by NES.

A database system keeps large data volumes and provides functionalities for access control, data consistency, fault tolerance and efficient data recovery. Furthermore, in a database it is possible to store the relationships between different types of data from different experiments, allowing for more sophisticated data analysis which are especially valuable to support research in multidisciplinary domains.

The NES Web interface and modular format provide an intuitive use of its data management functionalities and do not depend on any specific knowledge on informatics. NES was developed using open technologies and tools— such as the Django web framework and the PostgreSQL database management system—which can be easily installed and used in any research laboratory. Moreover, these tools make it capable of

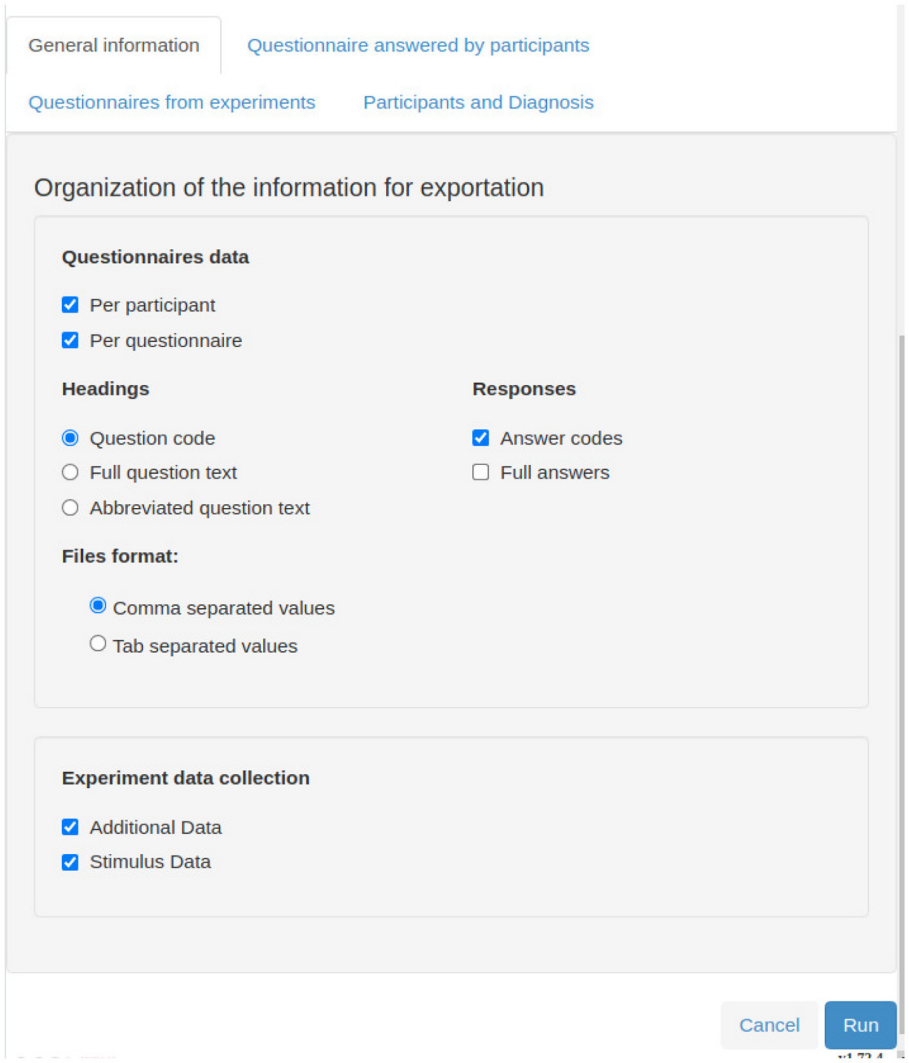


FIGURE 12 | NES graphical interface for selecting data export options.

supporting a large number of simultaneous users and handling large amounts of data. All the structured data managed by NES is stored in PostgreSQL. This includes the participant records, the description of the experimental protocols, the equipment settings, the research organization data, among others. This kind of data is efficiently handled with PostgreSQL.

All the files uploaded by users in NES, e.g., the EEG and EMG recordings and other types of data collections, are stored in the file-server system to facilitate their manipulation. This approach also enables the use of a distributed file system, a network storage device, or a cloud file storage to have storage scalability in a transparent way for NES.

NES is being used to manage experimental data of studies conducted in the NeuroMat research center. In particular, it has been used to construct an open database with data from TBPI studies. This initiative may allow the identification of functional markers related to the patients clinical improvement and foster the development of new investigative tools to unveil its mechanisms. Moreover, it aims at reducing the distance between clinical and experimental practice and encourage data sharing and reuse.

4.1. Limitations and Future Directions

NES provides basic functionalities for the registration of medical records because these data are required in several kinds of neuroscience experiments. However, it is important to emphasize that NES is not an electronic data capture (EDC) nor an electronic medical record (EMR) system and was not designed for these purposes. NES is not able to register information of nonhuman subjects. But the `Participant` data model can be easily extended to accommodate this type of data. Examples of attributes that should be considered for nonhuman subjects are identification, age, species, sex, stock or strain, house, and genetic characterization.

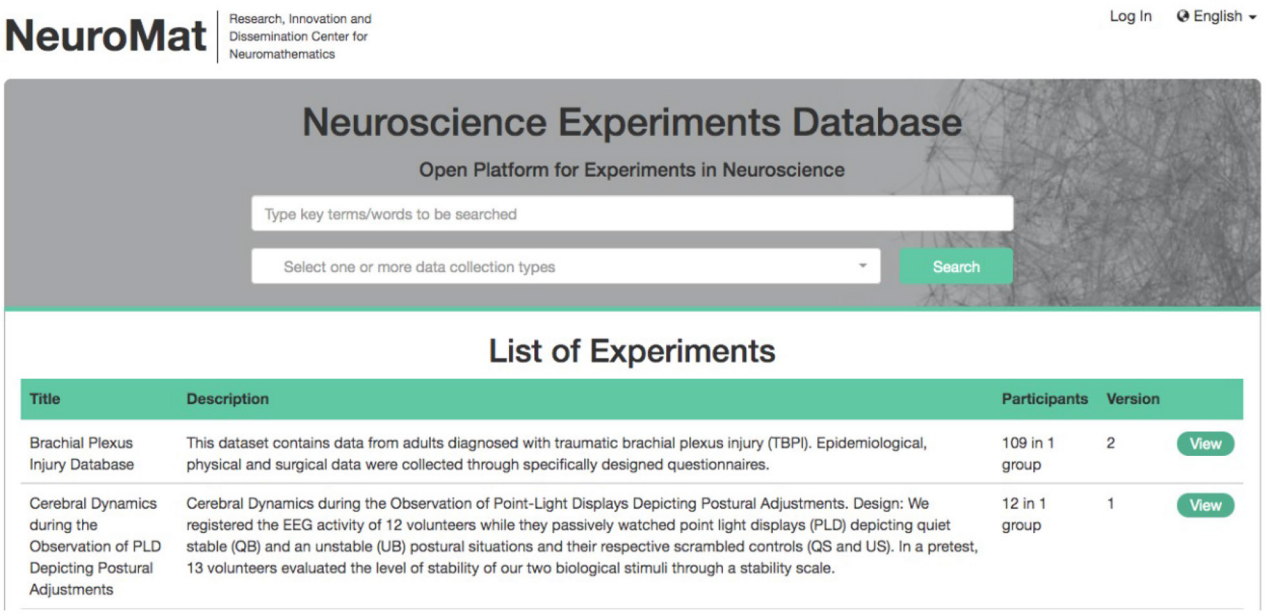


FIGURE 13 | NeuroMat Open Database (NeuroMat DB) home page.

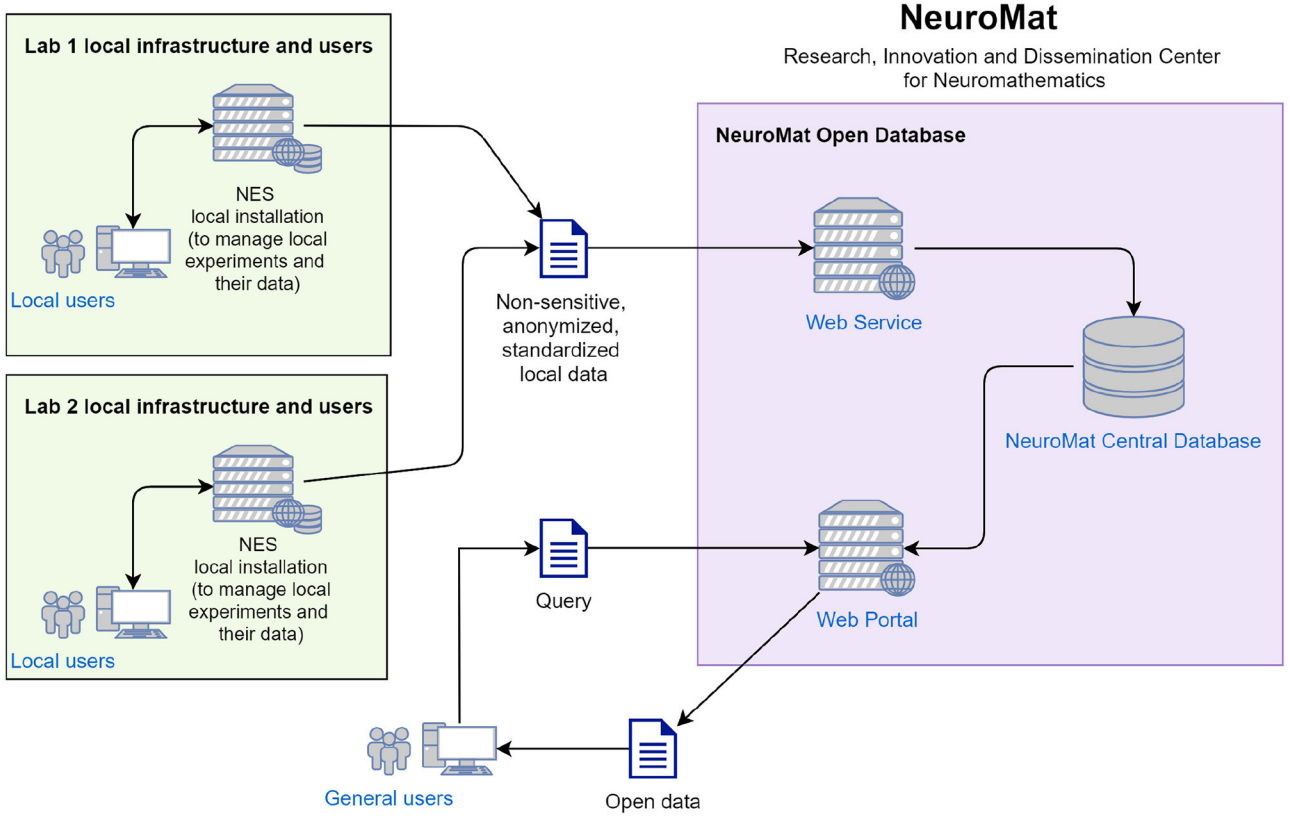


FIGURE 14 | Data capture with NES and public sharing in the NeuroMat Open Database.

NES has some simple features for data search and filtering. For instance, one can search for a given participant in a given study and perform basic filtering in the exportation module. Also, data filtering can be employed upon data transference to the NeuroMat Open Database (NeuroMat DB) portal. The portal supports data search using keywords via an Elasticsearch mechanism. To improve the tool with more advanced querying features, a new module for Data Searching and Visualization which will index all the data stored in the NES database is being designed.

NES has special functionalities that facilitate the management of electrophysiological data and metadata (i.e., EEG and EMG). However, currently it has limited support for experiments involving neuroimaging. For example, NES can read several EEG data formats and extract metadata from them, but it does not have an equivalent functionality to handle MRI and fMRI data. The extension of NES with a neuroimaging module will be implemented in the context of a scientific collaboration recently established with researchers from the Polytechnic Faculty of the National University of Asunción. In order to provide scalable storage for neuroimages, NES will be transformed into a cloud native system. This cooperative project also foresees the deploying of NES in research laboratories of the Neurology service of the Central Hospital of the Social Security Institute, in Asunción, Paraguay, to support studies of neurological disorders.

AUTHOR CONTRIBUTIONS

KB and CV conceived the original concept for the software. MR-O and ES contributed to the software implementation. MR-O and KB wrote the first draft of the manuscript. ER drew the conceptual data schema diagrams. All authors contributed to the data model, software design, revised the manuscript, and approved the submitted version.

FUNDING

This work was supported from research activity conducted as part of the Research, Innovation and Dissemination Center for Neuromathematics (NeuroMat) and funded by the São Paulo Research Foundation-FAPESP (No. 2013/07699-0), Brazilian National Council for Scientific and Technological Development-CNPq (No. 426579/2016-0 and 309560/2017-9), Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro-FAPERJ (No. E26/010002474/2016, CNE 202.785/2018, and E-26/010.002418/2019), Financiadora de Estudos e Projetos-FINEP (PROINFRA HOSPITALAR No. 18.569-8), and Paraguayan National Council of Science and Technology-CONACYT with support from FEEI (No. PINV18-665).

ACKNOWLEDGMENTS

We would like to thank Carlos Eduardo Ribas and Cassiano Reinert Novais dos Santos who were among the main developers of the NES code. We also thank Bia Santos, Cristiane Patroclo, Fernanda Torres, Juliana Maia, Lidiane Souza, and Maria Luiza Rangel, from the Laboratories of Neurobiology of Movement and Neuroscience and Rehabilitation of the Federal University of Rio de Janeiro, for their support in the requirement analysis and software testing.

REFERENCES

Abrams, M. B., Bjaalie, J. G., Das, S., Egan, G. F., Ghosh, S. S., Goscinski, W. J., et al. (2021). A standards organization for open and fair neuroscience: the international neuroinformatics coordinating facility. *Neuroinformatics* 1–12. doi: 10.1007/s12021-021-09522-x

Barkhof, F. (2012). Making better use of our brain MRI research data. *Eur. Radiol.* 22, 1395–1396. doi: 10.1007/s00330-012-2408-3

Fowler, D., Barratt, J., and Walsh, P. (2017). Frictionless data: making research data quality visible. *Int. J. Digital Curation* 12, 274–285. doi: 10.2218/ijdc.v12i2.577

Frishkoff, G., Sydes, J., Mueller, K., Frank, R., Curran, T., Connolly, J., et al. (2011). Minimal information for neural electromagnetic ontologies (MINEMO): a standards-compliant method for analysis and integration of event-related potentials (ERP) data. *Stand. Genomic Sci.* 5, 211. doi: 10.4056/sigs.2025347

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

Gibson, F., Overton, P. G., Smulders, T. V., Schultz, S. R., Eglen, S. J., Ingram, C. D., et al. (2008). Minimum information about a neuroscience investigation (MINI): electrophysiology. *Nat. Prec.* 1–7. doi: 10.1038/npre.2009.1720.2

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

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

Hernández, N., Duarte, A., Ost, G., Fraiman, R., Galves, A., and Vargas, C. D. (2021). Retrieving the structure of probabilistic sequences of auditory stimuli from eeg data. *Sci. Rep.* 11, 1–15. doi: 10.1038/s41598-021-83119-x

Koslow, S. H. (2000). Should the neuroscience community make a paradigm shift to sharing primary data? *Nat. Neurosci.* 3, 863–865. doi: 10.1038/78760

Krause, F., and Lindemann, O. (2014). Expyriment: a python library for cognitive and neuroscientific experiments. *Behav. Res. Methods* 46, 416–428. doi: 10.3758/s13428-013-0390-6

Martins, E. F., Lemos, T., Saunier, G., Pozzo, T., Fraiman, D., and Vargas, C. D. (2017). Cerebral dynamics during the observation of point-light displays depicting postural adjustments. *Front. Hum. Neurosci.* 11:217. doi: 10.3389/fnhum.2017.00217

Mathôt, S., Schreij, D., and Theeuwes, J. (2012). OpenSesame: an open-source, graphical experiment builder for the social sciences. *Behav. Res. Methods* 44, 314–324. doi: 10.3758/s13428-011-0168-7

Mouček, R., Brůha, 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

Patroclo, C. B., Ramalho, B. L., Maia, J. S., Rangel, M. L., Torres, F. F., Souza, L., et al. (2019). A public database on traumatic brachial plexus injury. *bioRxiv*, 399824. doi: 10.1101/399824

Peirce, J. W. (2007). Psychopy psychophysics software in python. *J. Neurosci. Methods* 162, 8–13. doi: 10.1016/j.jneumeth.2006.11.017

Peirce, J. W. (2009). Generating stimuli for neuroscience using psychopy. *Front. Neuroinform.* 2:10. doi: 10.3389/neuro.11.010.2008

Peschanski, J. A., dos Santos, C. R. N., and Ribas, C. E. (2020). Adequação de protocolos de exportação de um sistema de gestão de experimentos neurocientíficos às especificações de dados sem atrito. *Rev. IInovação Projetos e Tecnol.* 8, 83–96. doi: 10.5585/iptec.v8i1.16783

Poldrack, R. A., Fletcher, P. C., Henson, R. N., Worsley, K. J., Brett, M., and Nichols, T. E. (2008). Guidelines for reporting an fMRI study. *Neuroimage* 40, 409–414. doi: 10.1016/j.neuroimage.2007.11.048

Ramalho, B. L., Rangel, M. L., Schmaedeke, A. C., Erthal, F. S., and Vargas, C. D. (2019). Unilateral brachial plexus lesion impairs bilateral touch threshold. *Front. Neurol.* 10:872. doi: 10.3389/fneur.2019.00872

Ruiz-Olazar, M., Rocha, E. S., Rabaça, S. S., Ribas, C. E., Nascimento, A. S., and Braghetto, K. R. (2016). "A review of guidelines and models for representation of provenance information from neuroscience experiments," in *International Provenance and Annotation Workshop* (McLean, VA, USA: Springer), 222–225.

Sobolev, A., Stoewer, A., Pereira, M., Kellner, C. J., Garbers, C., Rautenberg, P. L., 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

Stoewer, A., Kellner, C. J., Benda, J., Wachtler, T., and Grewe, J. (2014). File format and library for neuroscience data and metadata. *Front. Neuroinform.* 8:15. doi: 10.3389/fninf.2020.00027

Sullivan, J. A. (2009). The multiplicity of experimental protocols: a challenge to reductionist and non-reductionist models of the unity of neuroscience. *Synthese* 167, 511. doi: 10.1007/s11229-008-9389-4

Teeters, J. L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., et al. (2015). Neurodata without borders: creating a common data format for neurophysiology. *Neuron* 88, 629–634. doi: 10.1016/j.neuron.2015.10.025

Torres, F. F., Ramalho, B. L., Patroclo, C. B., Souza, L., Guimaraes, F., Martins, J. V., et al. (2019). "Plasticity in the brain after a traumatic brachial plexus injury in adults," in *Treatment of Brachial Plexus Injuries, Chapter 3*, eds V. Vanaclocha, and N. Siz-Sapena (Rijeka: IntechOpen), 27–42.

Conflict of Interest: 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Neuroinformatics

METHODS
published: 07 January 2022
doi: 10.3389/fninf.2021.773890

BrainQuake: An Open-Source Python Toolbox for the Stereoelectroencephalography Spatiotemporal Analysis

*Fang Cai[1], Kang Wang[1], Tong Zhao[1], Haixiang Wang[2], Wenjing Zhou[2] and Bo Hong[1]**

[1] Department of Biomedical Engineering, School of Medicine, Tsinghua University, Beijing, China, [2] Epilepsy Center, Yuquan Hospital, Tsinghua University, Beijing, China

Intracranial stereoelectroencephalography (SEEG) is broadly used in the presurgical evaluation of intractable epilepsy, due to its high temporal resolution in neural activity recording and high spatial resolution within suspected epileptogenic zones. Neurosurgeons or technicians face the challenge of conducting a workflow of post-processing operations with the multimodal data (e.g., MRI, CT, and EEG) after the implantation surgery, such as brain surface reconstruction, electrode contact localization, and SEEG data analysis. Several software or toolboxes have been developed to take one or more steps in the workflow but without an end-to-end solution. In this study, we introduced BrainQuake, an open-source Python software for the SEEG spatiotemporal analysis, integrating modules and pipelines in surface reconstruction, electrode localization, seizure onset zone (SOZ) prediction based on ictal and interictal SEEG analysis, and final visualizations, each of which is highly automated with a user-friendly graphical user interface (GUI). BrainQuake also supports remote communications with a public server, which is facilitated with automated and standardized preprocessing pipelines, high-performance computing power, and data curation management to provide a time-saving and compatible platform for neurosurgeons and researchers.

Keywords: epilepsy, stereoelectroencephalography, electrode localization, Epileptogenicity Index, interictal high-frequency oscillation, Hough Transform

OPEN ACCESS

Edited by:
Ting Zhao,
Janelia Research Campus,
United States

Reviewed by:
Dingguo Zhang,
University of Bath, United Kingdom
Shuang Wang,
Zhejiang University, China

****Correspondence:***
Bo Hong
hongbo@tsinghua.edu.cn

Received: *10 September 2021*
Accepted: *26 November 2021*
Published: *07 January 2022*

Citation:
Cai F, Wang K, Zhao T, Wang H, Zhou W and Hong B (2022) BrainQuake: An Open-Source Python Toolbox for the Stereoelectroencephalography Spatiotemporal Analysis. Front. Neuroinform. 15:773890. doi: 10.3389/fninf.2021.773890

INTRODUCTION

Nearly 30% of the patients with epilepsy eventually become intractable patients resistant to antiepileptic drugs (Kwan and Brodie, 2000). To these patients, the intracranial stereoelectroencephalography (SEEG) surgery, first developed by Talairach and Bancaud at the Hospital Sainte Anne, Paris (Bancaud et al., 1965), is now a common clinical approach to consider about. SEEG aims at identifying the epileptogenic zones (EZs; Rosenow and Lüders, 2001) in the suspicious area of the brain of an individual by implanting depth electrodes and capturing the abnormal neural activities, followed by a resection or thermocoagulation surgery (Cossu et al., 2015; Wang et al., 2020). During this procedure, a large number of neurodata with multiple modalities occur. Presurgical MRI T1 structural image and CT image after the implantation surgery can, respectively, be taken as information for brain surface reconstruction and SEEG electrode localization (Behrens et al., 1994; Dykstra et al., 2012). Neural activities before

the resection surgery are recorded with SEEG electrodes for EZ localization and lesion analysis, usually lasting for 2 weeks. The neural activity acquired during the 2-week SEEG recording is vital to the presurgical planning (Cossu et al., 2015) and of great value to brain research (Zhang et al., 2019; Akkol et al., 2021). However, exploiting the large number of multimodal neurodata and managing them effectively remain a problem to be solved.

The SEEG electrode localization procedure using co-registered MR and CT images provides neurosurgeons with accurate anatomical positions of the implanted electrode contacts (Dykstra et al., 2012). The traditional and broadly used method of electrode contact localization mostly depends on visual checking and manual operations (Darcey and Roberts, 2010). After the registration of MR and CT images, technicians view the CT image slice by slice, locating highlighted contact voxels and mapping the positions onto the MRI (Darcey and Roberts, 2010). Trouble occurs since every patient may have 100 contacts implanted on average, and one should check the slices back and forth for a highlighted contact centroid, which is a complicated and time-consuming task. Several previous studies have proposed semiautomated methods (Blenkmann et al., 2017; Hamilton et al., 2017; Narizzano et al., 2017; Qin et al., 2017; Li et al., 2019) to improve the effectiveness and precision of electrode contact localization. The SEEG Assistant (SEEGA) extension of the 3D Slicer applies an algorithm of the center-of-mass convergence for the contact segmentation step (Arnulfo et al., 2015; Narizzano et al., 2017), which shows great feasibility and robustness in locating contacts along each electrode shaft. However, this method requires a prior manually defined fiducial file of the planned starting and ending points of each electrode and an additional presurgical CT scanning. Another study (Qin et al., 2017) inherits the convergence algorithm and develops a preprocessing workflow to reduce the required input. This workflow includes MRI and CT registration, masking, eroding, and clustering steps but still needs to insert several pause points for visual checking and manual adjustments. Another toolbox (Blenkmann et al., 2017) implements a k-means clustering algorithm to segment contacts along each electrode, in which the voxels of each electrode should be carefully thresholded, otherwise the contacts may not be completely segmented.

In the clinical SEEG data analysis, doctors are mainly concerned about the effect of a few episodes of ictal data for the location of EZs. Channels with relatively early abnormal activity during the seizure often indicate the potential EZs. A previous study defined an Epileptogenicity Index (EI) using the onset of high-frequency energy to predict the onset area (Bartolomei et al., 2008). However, in some cases, the onset period may not be captured to provide sufficient diagnostic information. In contrast to only a few seizures during the monitoring period, most of the SEEG signals recorded are seemingly ordinary interictal data. The sporadic abnormal activities in the interictal interval, such as spikes or high-frequency oscillations (HFOs), can be used as plausible pathological markers of EZs. Because the intracranial EEG recording consumes huge storage space, recording an 80-channel intracranial EEG at a sampling rate of 2,000 Hz for 24 h may generate a data volume of about 50 GB. It is time-consuming for surgeons to extract sparse interictal pathological activities from the long-term SEEG. Currently, the interictal data cannot be fully and effectively traversed by surgeons and thus is usually deleted. The value of the interictal data is mostly underestimated. Therefore, there is an urgent need to detect abnormal activities in interictal SEEG data to extract pathological information and reduce the workload of clinicians. Both HFO activities (Navarrete et al., 2016) and spike detection algorithms (Barkmeier et al., 2012) have been developed based on waveform morphology, but indexation methods that efficiently extract interictal epileptic discharge events are yet to be developed. In addition, the performance of current interictal event detection methods heavily depends on the manual selection of the parameters (Remakanthakurup Sindhu et al., 2020). Our interictal data analysis module is designed to minimize manual interference by implementing an automatic HFO detection method and retaining only necessary parameter settings such as filter ranges and channel selections.

After electrode localization and data analysis, cortical surface reconstruction is an essential step for better visualization. Several previous studies have developed the reconstruction procedure (Dale et al., 1999; Fischl, 2012; Henschel et al., 2020; Zöllei et al., 2020). FreeSurfer group releases tools and pipelines publicly (Fischl, 2012). They built a reconstruction pipeline, "recon-all," covering from primary operations such as motion correction and skull-stripping, to final steps such as segmentation and cortical parcellation. Several subsequent studies have also proposed advanced reconstruction tools such as specifically, "infant-FreeSurfer" (Zöllei et al., 2020) for covering all ages of subjects and "FastSurfer" deep learning pipeline (Henschel et al., 2020) for solving the time-consuming problem. However, FreeSurfer software and its advanced tools can only be executed on Linux-based operating systems (OS). Virtual machine configuration and the usage of terminal lines can be troublesome for some Windows users. Moreover, there is often a lack of local computing power for rapid surface reconstruction in the clinical setting.

In this study, we present BrainQuake, an open-source Python software, providing epilepsy surgeons with tools and integrated pipelines of surface reconstruction, electrode contact localization, and ictal and interictal SEEG analysis for presurgical evaluations. The integration aims at automatically executing the whole workflow with fewer input files and fewer pause points. BrainQuake is designed as an end-to-end, highly automated, time-saving software, free to be downloaded and compatible with both Linux and Windows OS. With a comprehensive data processing platform established, surgeons can take the most advantage of neurodata and make reliable presurgical evaluations for those epilepsy patients. We hope this software can be helpful to clinical practice and human neuroscience studies using SEEG.

MATERIALS AND REQUIREMENTS

Software Overview

BrainQuake is an open-source Python software for image and SEEG data processing of refractory epilepsy patients. BrainQuake consists of four modules, namely, surface module, electrode module, ictal module, and interictal modules (**Figure 1**). The surface module is used for surface reconstruction of the MRI

T1 image of the patient. We incorporated a GUI, a client-server communication mode, a public server with powerful graphics processing units (GPUs), and a data curation system, to ensure that users share a time-saving, private, and stable data preprocessing pipeline. The electrode module consists of a pipeline to locate and anatomically label the SEEG electrode contacts using both preoperative T1 image and postoperative CT image. The ictal module and interictal module analyzed the recorded SEEG data and then pinpoint the suspicious seizure onset zones (SOZs) using EI and High-Frequency Events Index (HI), respectively. Finally, BrainQuake provides a comprehensive visualization result of the 3D brain surface of an individualized patient, with SEEG contacts and SOZ predictions projected on it. We developed GUIs for all these modules (**Figure 2**), and tutorials can be found along with installation packages.

Data

Subjects

The SEEG electrodes, or intracranial depth electrodes, were used in human subjects undergoing epilepsy surgical treatment. We analyzed the data from five patients temporarily implanted with SEEG electrodes (8–16 contacts per electrode, 2 mm diameter, and 3.5-mm center-to-center spacing). Intracranial EEG was continuously recorded for 2 weeks on average, and MRI and CT images were, respectively, acquired before and after the implantation operation. The surgeries were conducted in the Department of Neurosurgery and Epilepsy Center, Tsinghua Yuquan Hospital. Data collection and scientific workup were approved by its Institutional Review Board.

Example Data

We provided eight sets of sample data so that potential users can follow the data format and file structure and go through the procedures in BrainQuake. Sample data are available at https://doi.org/10.5281/zenodo.5675459, such as MRI T1 image, CT image in NIfTI-1 type, and recordings of ictal and interictal EEG data (up to 2 h per patient) for each sample. The file structure is shown in **Figure 3**. FreeSurfer "recon-all" results are also included since we used some of their intermediate files (mri/orig.mgz, brainmask.mgz; surf/lh.pial, rh.pial) in our modules. Two separate directories, namely, BrainQuake dataset and FreeSurfer dataset, will be configured during the initialization of the software.

Operating Requirements

The codes are divided into the client part and the server part. Computers running either Linux, Mac OS X, or Windows should run the client Python GUI code. For the server part, it should be running on Linux or Mac OS X, since FreeSurfer works only on Linux. We recommended users install the client GUI code and communicate with a public server we provided and leave all the time-consuming works (e.g., surface reconstruction, CT and MRI image registration) to it. Essential processed data for functional modules in BrainQuake can be downloaded from the server. If facilitated with a Linux-based server at local, one can still download and install the server codes and run the whole pipeline within their own workspace. On the remote server side, FreeSurfer (version 6 or higher) should be properly installed as well as the packages mentioned previously. Full installation tutorials can be found on https://github.com/HongLabTHU/Brainquake. Detailed operating requirements are listed as follows:

1. Computers running on Linux, Mac OS X, and Windows should run the client codes (i.e., Python scripts outside the "Server_codes" folder on the GitHub of BrainQuake).
2. Server codes should be run on a Linux-based server, with FreeSurfer (version 6 or higher) installed.
3. Processor speed: 2.0 GHz or higher recommended.
4. RAM: 8 GB or higher recommended.
5. Python version: 3.6 or higher.
6. Third-party dependencies: numpy, nibabel, matplotlib, scikit-learn, scipy, mne, vtk, and mayavi.

The public server [Ubuntu 18.04, 40 central processing units (CPUs), 2.10 GHz] we provided assigns eight cores to each "recon-all" task for parallel computing and can hold up to three tasks running simultaneously. Each "recon-all" task lasts 3 h on average. Server codes are also provided on the GitHub of BrainQuake so that one can facilitate their own server for reference. The output package of a surface reconstruction task from the server pipeline of BrainQuake includes a typical reconstruction result folder (produced by FreeSurfer), an "orig.nii.gz" file (produced by FreeSurfer command "mri_convert"), a "mask.mgz" file (produced by FreeSurfer command "mri_binarize"), and a registered "<name>_CT_Reg.nii.gz" file (produced by FSL command "flirt" with "orig.nii.gz" as its reference image). Producing all of these files and folders requires FreeSurfer installed in the operating environment, so if a potential user prefers not to apply the client-server mode, one can always import their own "recon-all" folders with all these Supplementary Files prepared.

METHODS

Image Processing Modules

Surface Module

FreeSurfer provides a complete pipeline, "recon-all," for surface reconstruction, which is compiled with abundant tools such as skull-stripping, image registration, cortical reconstruction, and segmentation. More time-saving or specific pipelines such as "FastSurfer" (Henschel et al., 2020) and "infant-FreeSurfer" (Zöllei et al., 2020) have been released in recent years. We integrated all those pipelines in the provided server and also provided processing options in the surface module GUI so that users no longer need to deal with the terminal when using "recon-all" or wait too long for a reconstruction result since the server is facilitated with GPUs and the average processing time is 3.5 h for "recon-all" and only 30 min for "FastSurfer" and "infant-FreeSurfer." Windows users need not configure a virtual machine for installing FreeSurfer locally since our server can undertake all the preprocessing works.

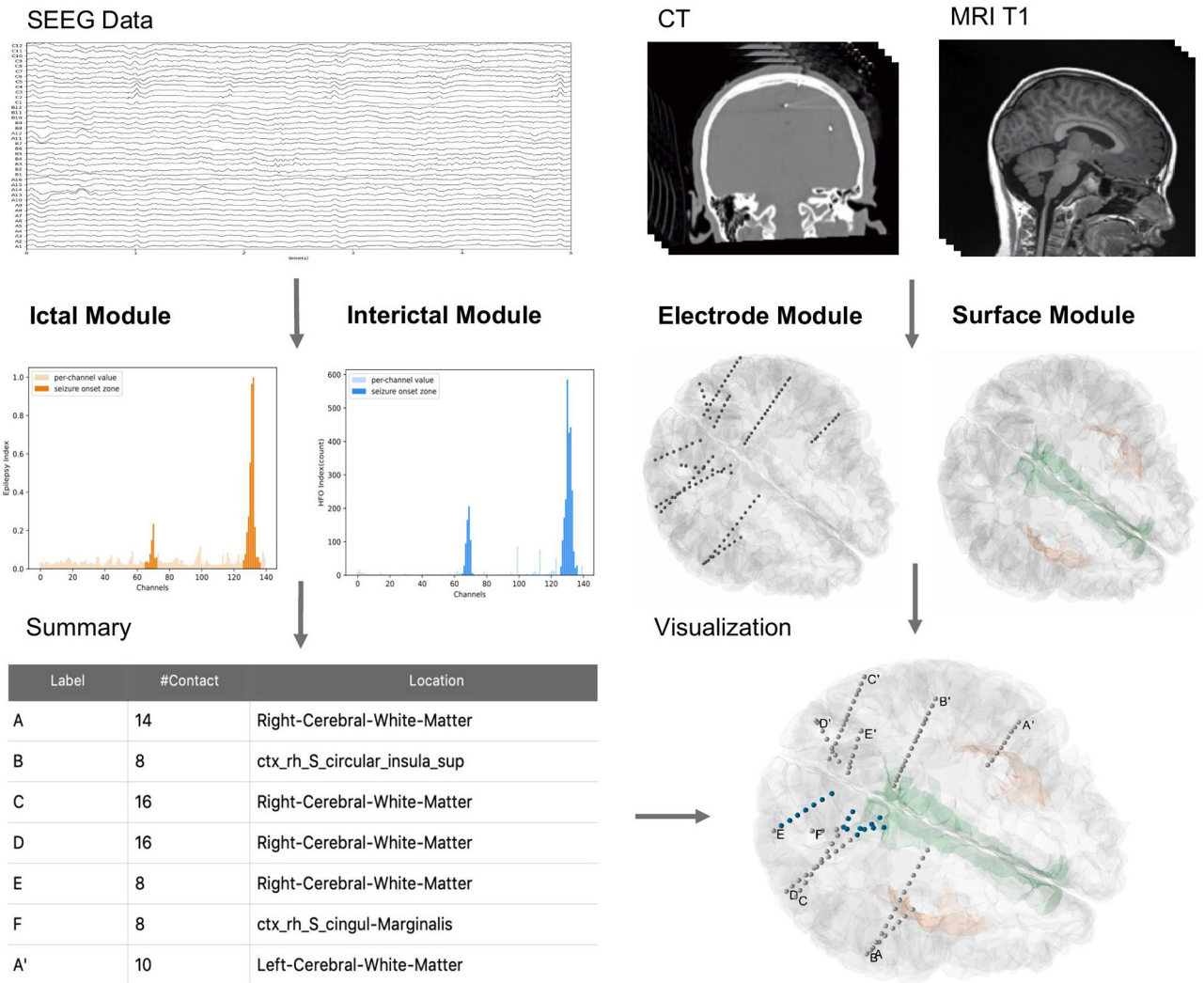


FIGURE 1 | General overview of BrainQuake structure. BrainQuake is designed to analyze the SEEG data (ictal and interictal), CT images, and MRI T1 images. Ictal and interictal modules are used to predict suspect contacts within seizure onset zones (SOZs). The electrode module exploits the graphic information from a CT image to locate the stereoelectroencephalography (SEEG) electrodes and contacts, as well as project them onto the brain surface, which is reconstructed by the surface module. The locations of suspect contacts are marked (blue) on the 3D plot of the surface and electrodes, giving a brief overview of the presurgical evaluation results.

Electrode Module

Either processed manually or semiautomatically, the main idea of electrode contact segmentation is to identify the brightest voxels in a CT image as contact positions along each depth electrode. To conduct an autonomous pipeline of contact segmentation, we should make the best use of the image properties. The electrode module of BrainQuake requires the input data of only a postsurgical CT NIfTI image and a result package of surface reconstruction. The pipeline in the module includes three parts, namely, image preprocessing, electrode clustering, and contact recognition (**Figure 4**).

Preprocessing

Before we could autonomously identify an electrode or contact, we must ensure that the image contains only the intracranial area of a brain since the skulls, teeth, or some electrode supports outside the brain are hard to be distinguished from the electrodes based on the voxel value difference of a CT image. In the preprocessing step, we registered the CT with the standardized MR image generated in the surface module. This registration step uses FSL "flirt" (Jenkinson et al., 2012) after surface reconstruction in the surface module. Then, the registered CT can be masked with a skull-stripped MR image in the surface data package to remove the extracranial part of the CT data since they are now in the same coordinate. At this time, the CT image contains only the information about the intracranial brain and the electrodes, the two of which show a significant difference in their voxel value ranges. Electrode voxels are much brighter in the image, so they can be extracted simply by thresholding (**Figure 4A**).

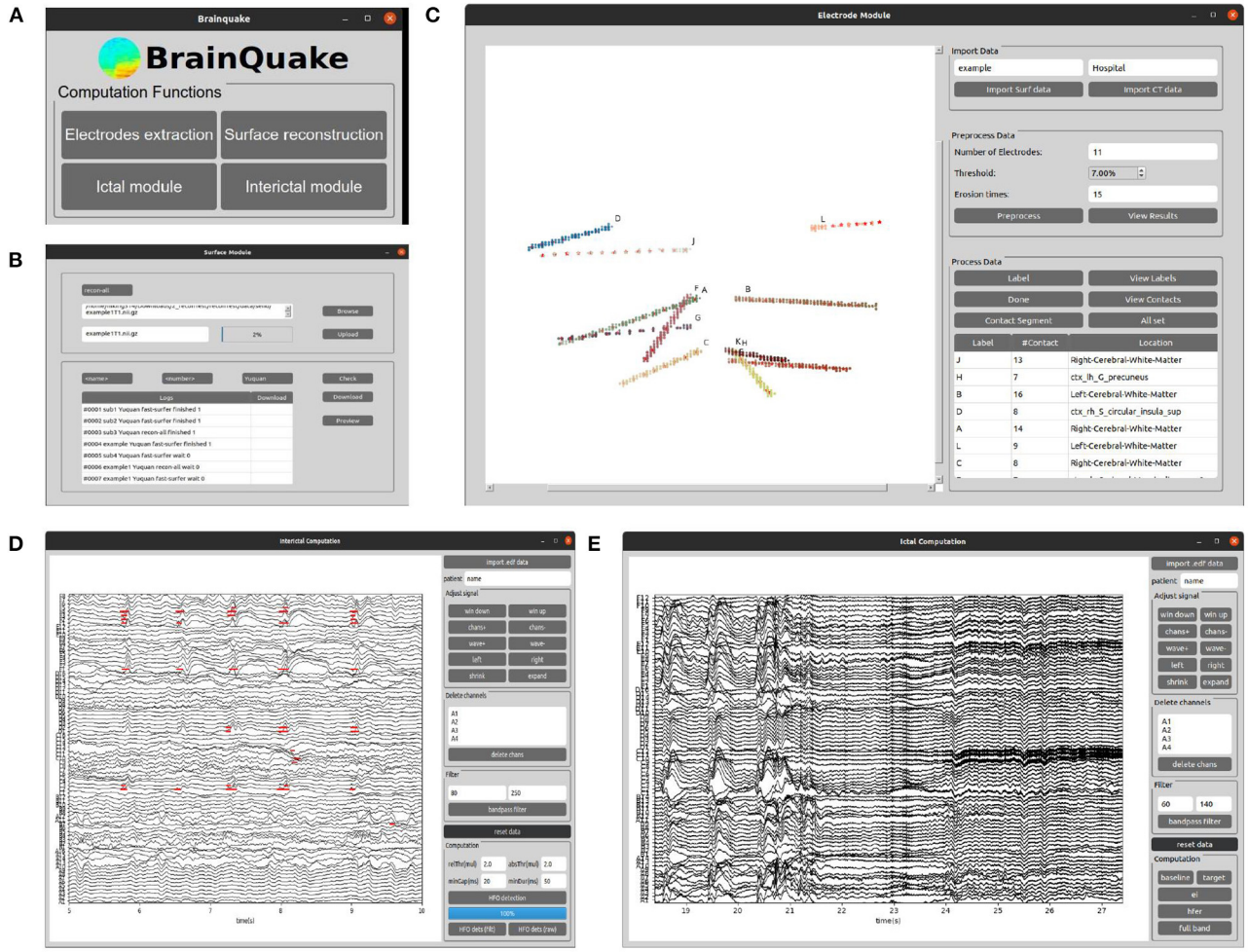


FIGURE 2 | The graphical user interfaces (GUIs) of the main window of BrainQuake and the four functional submodules. **(A)** BrainQuake main window; **(B)** Surface module; **(C)** Electrode module; **(D)** Interictal module; **(E)** Ictal module.

Hough Transform and Gaussian Mixture Model

After extracting the electrode voxels into point clouds (**Figure 4B**), we need to identify the number and axes of electrodes and label each voxel into different electrode clusters. This step is completed in most of the previous works by clustering algorithm with manual adjustment. In BrainQuake, we developed a method of combining 3D Hough Transform, a pattern recognition algorithm, and Gaussian Mixture Model, a clustering algorithm, to label voxels into different electrode clusters (**Figure 5**).

Normal clustering algorithms randomly pick some centroids in CT images, classify the voxels into clusters, and calculate the new centroid of each cluster. After multiple iterations, theoretically, voxels belonging to the same electrode can be assigned to the same cluster. However, the clustering algorithm is strongly dependent on the initial selection of centroids. With an improper initialization of the random centroids, the true distribution of electrode clusters can be difficult to estimate. There is a high probability that we would get a locally optimal clustering result, definitely requiring a manual intervention here to fix it, for example, to merge some of the clusters to form a real electrode or to split two or more electrodes in the same cluster.

Our method fixes this issue by adding a Hough Transform before clustering. Hough Transform is a common method used in computer vision or digital image processing (Illingworth and Kittler, 1988). It can be used to detect a certain class of shapes in an image automatically. The main idea of Hough Transform is that for a specific shape, we have chosen a set of parameters and created a parameter space. For example, the parameter we usually used to describe circles can be center and diameter, while the parameter of 2D lines can be slope and intercept. Suppose we have a raw image with a mixture of dots on it. Each dot will vote in the parameter space for every possible parameter set they can contribute. Positions in the space with the highest votes are recognized as the parameter sets describing the most obvious shape in the raw image. In our case, SEEG electrodes in a CT image are a combination of line-shaped objects in 3D space. The parameter space is established to represent the line direction

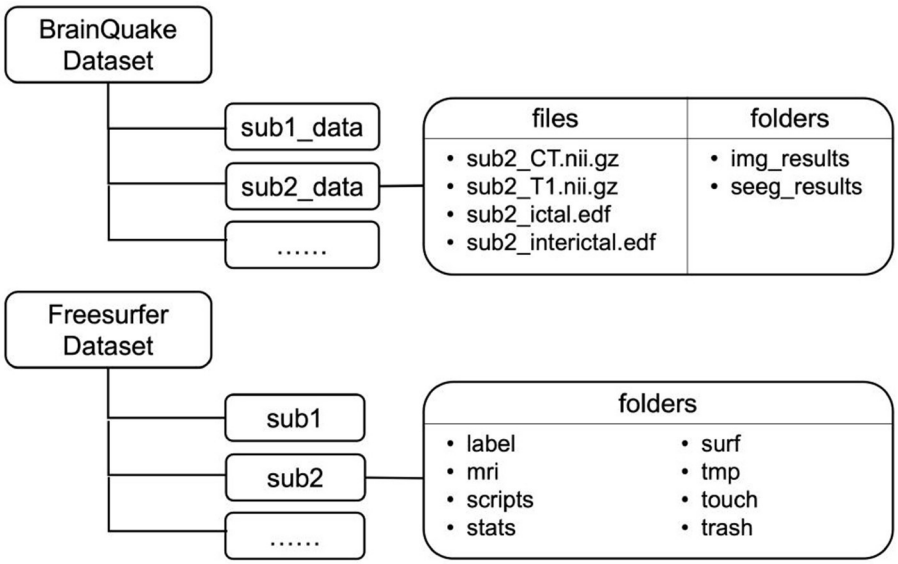


FIGURE 3 | File structures of two datasets implemented in BrainQuake. Temporary and final results are saved under the folders of each subject.

(horizontal orientation and altitude) and the distance between the coordinate origin and the line.

First, we transformed those voxels into point clouds (**Figure 5A**). Then, we applied a 3D line Hough Transform to detect line-shaped trajectories (Jeltsch et al., 2016; Dalitz et al., 2017) in the point clouds, returning centroid and axis direction of each electrode cluster. At this stage, we got a set of approximate but not precise results representing the position of each cluster (**Figure 5B**), which can be a good set of prior knowledge to start clustering. After that, we used the Gaussian Mixture Model (Reynolds, 2009; Pedregosa et al., 2011) to assign each point to the electrode cluster it belongs, since the point clouds can be viewed as a mixture of different line-shaped 3D Gaussian kernels (**Figure 5C**). After a successful clustering, the axes directions of electrodes can be regressed (Pedregosa et al., 2011). This combinatory method makes use of both electrode geometric prior and voxel distribution in a CT image, which shows excellent accuracy and robustness in our experiments.

Contact Segmentation

In the SEEG contact segmentation step, our general goal was to automatically recognize the relatively brightest voxels, which are viewed as contact positions, along each electrode shaft. We mainly divided the process into four sub-steps, namely, locating the head voxel, locating the target contact, stepping toward the next contact, and locating the rest contacts along the shaft.

First, we applied a linear regression (Pedregosa et al., 2011) to each electrode cluster of voxels to get the direction parameter (coefficients between x-y/y-z/z-x axes) of the electrode shaft track in the 3D space coordinate. We then used the direction to locate two voxels, respectively, to be the head and tail of the cluster. As a general assumption that the head voxel is always closer to the center of the brain (i.e., the center of the image space), we can locate the position of the head voxel, which is much close to the target contact.

Second, we applied a "center-of-mass" convergence algorithm (Arnulfo et al., 2015) to locate the target contact. We viewed each voxel value as the "mass" of a single voxel or "weight" of this point. In this way, the center-of-mass is defined as the "heaviest" point within a small region of voxels. After finding out the head voxel, we calculated the center-of-mass of its surrounding region (a geometry-restricted cubic volume with respect to the actual contact size, $2 \times 2 \times 2$ mm cube in our case). We then again calculated the next center-of-mass within the surroundings of the newly found center-of-mass. After 1–2 iterations of this procedure, the calculated center-of-mass eventually converges to the brightest voxel around the head of the electrode (i.e., the real target contact position).

Third, as we already knew the electrode track direction and the target contact, stepping out a specific distance along the direction from the target contact can give us a position close to the next contact. The step size should equal the real distance between two adjacent contacts (3.5 mm in our case). In this case, we made sure that the position found was close enough to the next contact, which was ready for another center-of-mass convergence procedure.

Finally, using the same center-of-mass convergence and the stepping strategy, the rest contacts can be recognized one by one. In this iterative process, we also set a geometrical restriction to ensure that the directed positions are always settled within the cluster by doubling the weights of the voxels in the cluster (**Figure 4D**).

Validation Method of Electrode Localization

We used two methods to validate the results of the electrode module, namely, visual inspection of the electrode positions and

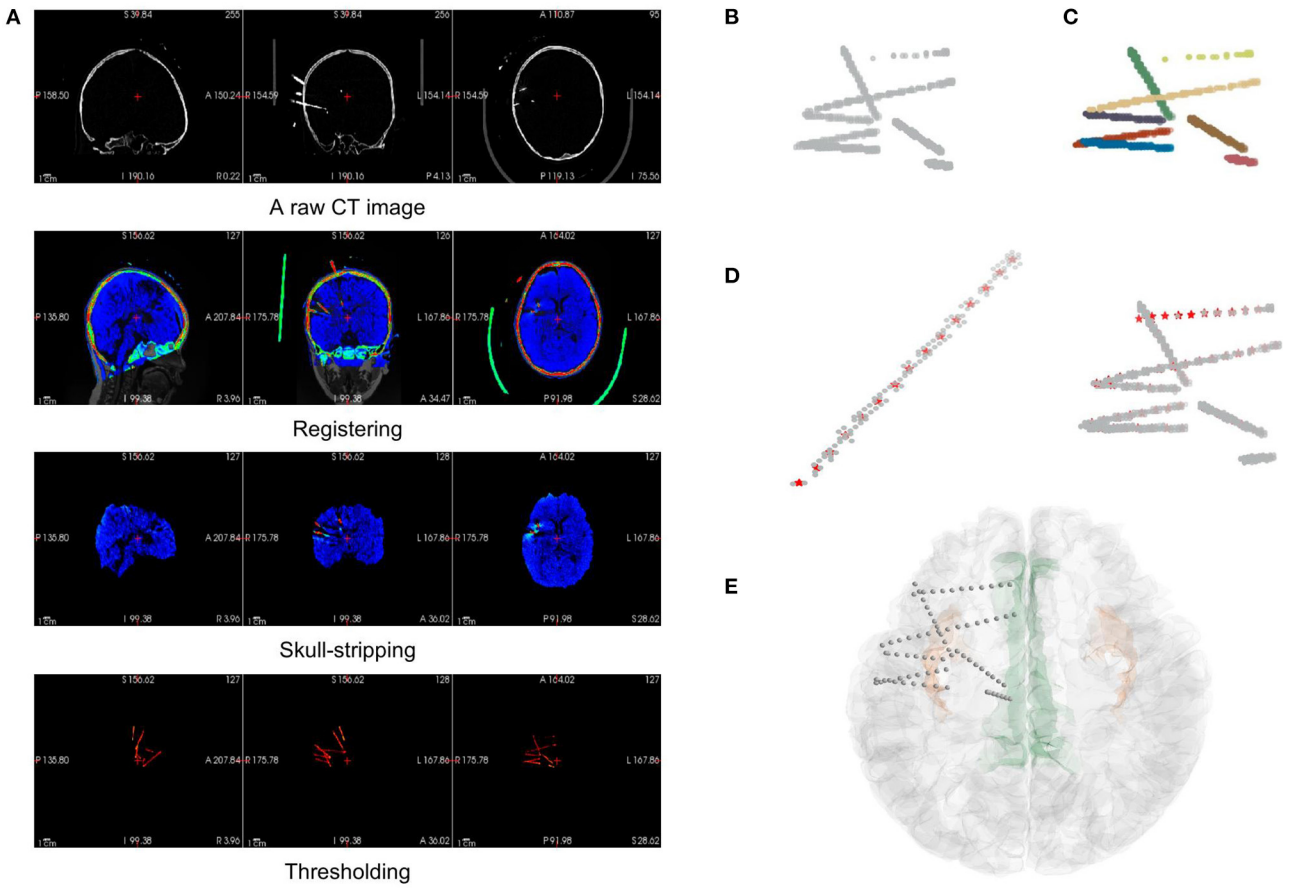


FIGURE 4 | The pipeline of electrode localization and contact segmentation procedures in the electrode module. **(A)** The preprocessing step includes image registration from the raw CT of a subject to MRI (orig.mgz after surface reconstruction), skull-stripping of registered CT (using brainmask.mgz after surface reconstruction), and thresholding of electrodes in the CT data. **(B)** The coordinates of electrode voxels in the CT image after thresholding can be extracted and plotted, viewing as a mix of point clouds. **(C)** After applying a Hough Transform and Gaussian Mixture Model algorithm, the electrodes are clustered and labeled by different colors. **(D)** Contact segmentation step: contact positions are recognized one by one by converging to the center-of-mass based on voxel values. Contact positions are marked as red asterisks. **(E)** The results of the contact segmentation pipeline are projected onto the 3D surface space.

quantitative measurements of the electrode contact distribution. The recognized contacts were projected onto the 2D slice of the fusion of MR and CT images. Then, we scanned through all these slices and visually checked if the electrodes and the highlighted electrode shaft on CT slices were overlapped.

To quantitatively estimate the accuracy of contact localization, we must define a gold standard of contact positions and then estimate the contact deviation error one by one. Usually, a group of clinical experts should be invited to view through all those image slices and mark the contact positions manually. However, due to the artifacts of each contact in the CT images, one may find it tough to segment those contacts since the adjacent contact pairs are usually merged. Thus, we could not trust the manual segmentation results as a gold standard. In this study, we estimated two indirect metrics, namely, axis-contact distance (i.e., distances between contacts and their estimated shaft axis) and adjacent contact distance of each adjacent contact pair (Arnulfo et al., 2015; Narizzano et al., 2017). Both of the metrics are based on the geometric properties of the SEEG electrodes. Contacts along the same electrode shaft are line-shaped regressed, and the axis-contact distance ideally can be close to 0 mm. The axis-contact distance is defined as the distance between the contact position and the regression line of the electrode shaft. It reveals how straight the contacts are located. The electrodes we used have a fixed spacing distance of 3.5 mm between neighboring contacts, so the adjacent contact distance we estimated should be distributed similarly to a Gaussian with a mean of 3.5 mm and a trivial variance as much as possible. However, it is often the case that the electrode shaft bends slightly and the contacts deviate from the line after the implantation surgery, which in some way causes these two distributions to be not so ideal (refer to the "Discussion" section).

SEEG Data Analysis Modules

Ictal Module

For ictal data, clinicians tend to mark the areas where the pathological activity occurs earlier as the potential SOZs. Based on this consensus, an EI method is commonly used to predict the

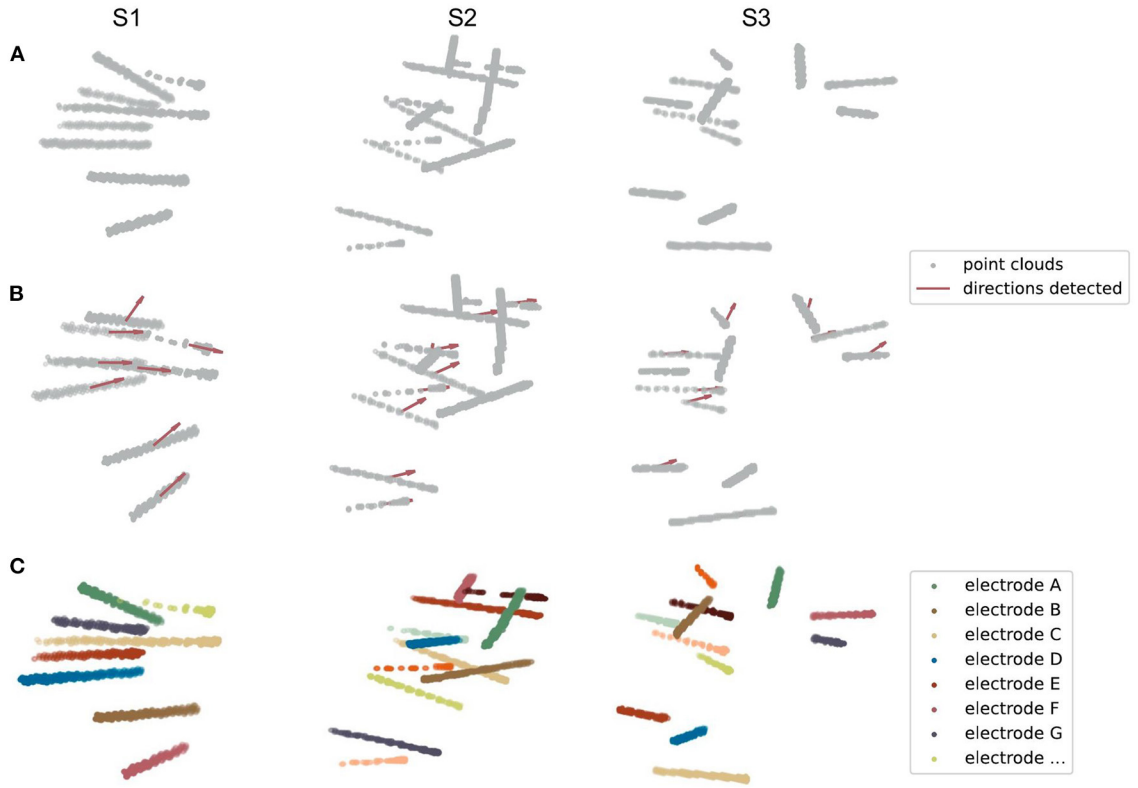


FIGURE 5 | Three examples of electrode point clouds have been 3D Hough-transformed and then clustered using the Gaussian Mixture Model. **(A)** The initial point clouds of electrodes are extracted from the CT intracranial image of an individual after several preprocessing steps. **(B)** The centroids and directions (showing by the red arrows) of SEEG electrodes are detected by the Hough Transform algorithm of a line in 3D coordinates. **(C)** The clustered electrodes are marked using different colors, after applying the Gaussian Mixture Model and the prior knowledge of the centroids and directions of clusters generated from **(B)**.

SOZs (Bartolomei et al., 2008). In this study, we implemented a simplified EI measurement in BrainQuake, predicting the SOZs by quantifying the combined effect of the timing order and the strength of high-gamma energy change in each channel during the onset process of the seizure (Zhao et al., 2019).

Before we did any automatic computation, we first filtered the raw signals into high-gamma frequency bands (60–140 Hz, power noise at 50 Hz) using a second-order IIR notch digital filter and a fifth-order Butterworth IIR filter (Virtanen et al., 2020). We then manually selected a segment of the baseline (BL) data, as well as a segment of the target data containing the initial onset process of seizure. The BL data should be located before the seizure onset, and a range of around 60 s should be enough for it. The target data should cover the seizure onset process, that is, to start somewhere before the onset and end within the seizure. The length of the target data is not limited as long as it covers the seizure onset process.

After the manual selection, we calculated an EI for each channel. First, the band-passed signals are transformed into a high-frequency energy spectrum by amplitude squaring and window smoothing (500-ms window length, 1 sample point per step). Second, we calculated the average value of the high-frequency energy of the BL data, which is used to normalize the high-frequency energy by division. In this way, we obtained the normalized high-frequency energy (NHFE) (**Figure 6A**). Third, a threshold of onset time was calculated for each channel *i*, which is 10 times the standard deviation (SD) of baseline (BL) NHFE above its maximum value as follows:

$$thre_i = \max\left(NHFE_{BL,\,i}\right) + 10\sigma\left(NHFE_{BL,\,i}\right)$$

For each channel, once the normalized energy in the target data exceeds its corresponding threshold, we decided this moment as the onset time of its abnormal activity. Fourth, we sorted the channels by their onset time and defined the time coefficient (TC) as the reciprocal of the order of each channel (i.e., 1, 1/2, and 1/3). Also, we calculated the average energy of each channel in a 250-ms period right after the earliest onset time as energy coefficient (EC) using the NHFE. Finally, the EI of each channel *i* is obtained by the following:

$$EI_i = \sqrt{TC_i \times EC_i}$$

As we can notice, EI, combining the effect of timing and energy strength, can be used to quantify the degree of epileptogenicity of each electrode channel (**Figure 6A**).

Interictal Module

A previous study on the SEEG interictal data found that both HFOs and spikes are the reliable biomarkers of SOZ, while HFO

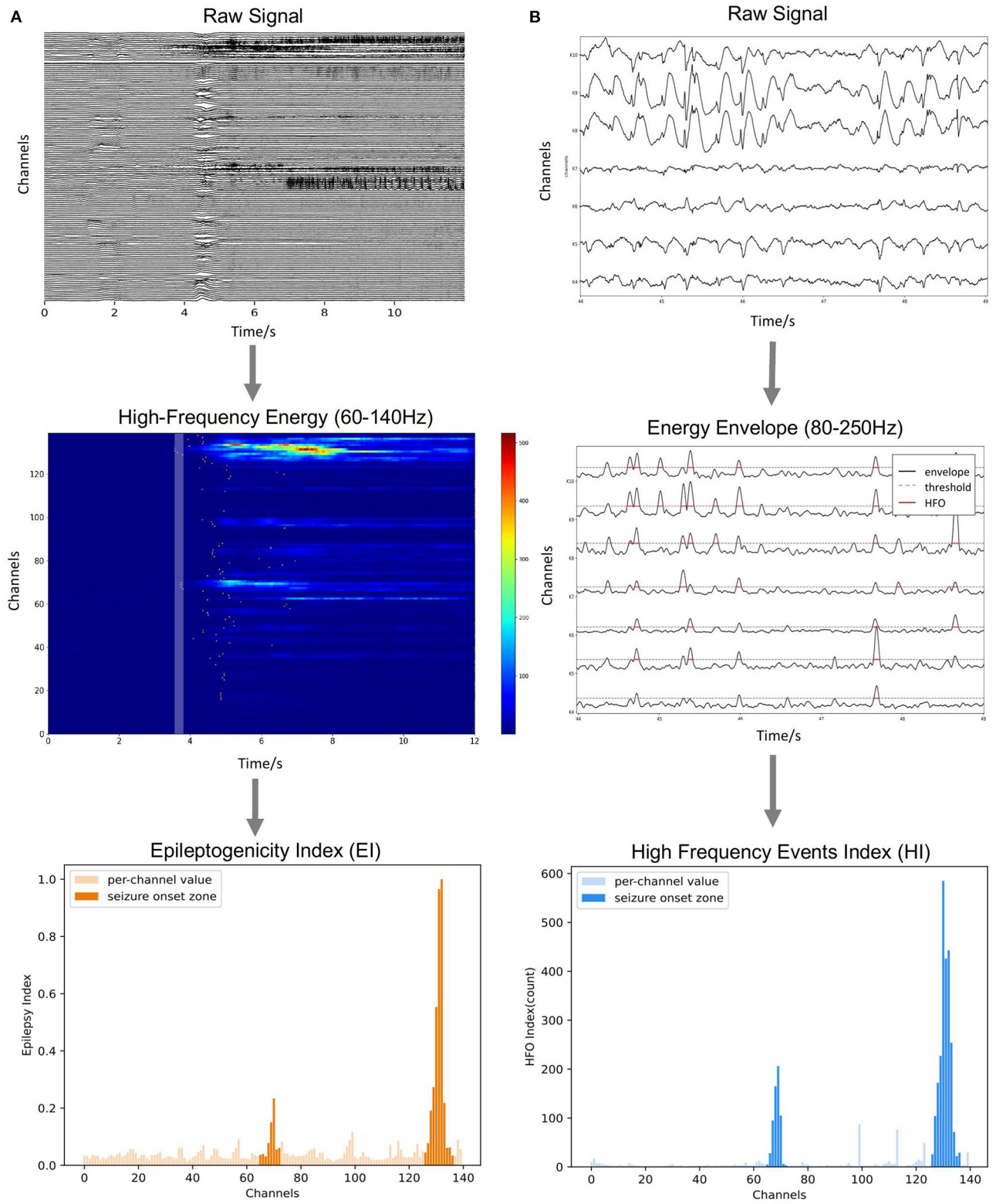


FIGURE 6 | Methods of ictal and interictal SEEG data analysis. **(A)** Onset timing order and energy strength during the initial stage of seizures are sorted to calculate the Epileptogenicity Index (EI). **(B)** Numbers of over-threshold high-frequency events are counted as High-Frequency Events Index (HI).

has better specificity for SOZ than spikes (Wang et al., 2017; Roehri and Bartolomei, 2019). The HFO subcategory, 80–250 Hz ripple component, is relatively more common than a higher frequency component (Wang et al., 2013). This frequency band can also take into account the spike activity, which is similar to a full-band signal (Roehri et al., 2017; Cai et al., 2021). Therefore, for the interictal data, we extracted the pathological activity by detecting the short-term abnormal energy enhancement in the 80–250 Hz band, providing an efficient indexation method through unified energy detection. Specifically, we used the Hilbert transform to extract the energy envelope in the 80–250 Hz band of the signal (i.e., users can adjust the frequency range for their own cases). The filter setting applied is a second-order IIR notch digital filter with a quality factor set to be 30, followed by a five-order Butterworth band-pass filter (Virtanen et al., 2020). We calculated the median value of the whole envelope (global, S_{global}) and the median value of each contact (local, S_i). Considering both of them, we set a synergistic threshold for each contact as follows:

$$thre_i = 2 \times \max(median\,(S_i)\,,\ median(S_{global}))$$

The time range where the envelop exceeds the threshold is marked as abnormal activity (**Figure 6B**). When the interval between two adjacent abnormal activities is too small (<20 ms), they are considered to belong to the same event and merged, and the abnormal activities of the very short duration (<50 ms) are excluded. Finally, the number of abnormal activities (HI) calculated for each channel is used as an index to measure the relative likelihood of each contact of being in the SOZ.

RESULTS AND VALIDATION

We processed all four functional modules using the MRI/CT images and the SEEG data acquired from 8 epilepsy patients. The time required for surface reconstruction was either around 0.5 h using FastSurfer or 3.5 h using FreeSurfer recon-all on the public server (40 cores, 2.1 GHz, 64 GB RAM). The preprocessing step in the electrode module for each subject is around 15 min, mostly spent on the image registrations of MRI and CT using the FSL "flirt" command. Contact localization consumes only 30 s for each subject on average. A 70-s interictal SEEG costs around 40 s for EI calculation, and the 2-h interictal data costs around 20 min for HI calculation.

Electrode Module Validation

We processed 74 electrodes with 743 contacts implanted in eight patients in total. During visual inspection, all 74 electrodes were perfectly matched with the highlighted electrode shaft artifacts on CT images (**Figures 7A,B**). For quantitative validation, we estimated two metrics, namely, axis-contact distance and adjacent contact distance error, to measure whether the distributions of recognized contacts obey the geometric rules of the SEEG electrode. In statistics, 95% of the contacts were <0.1 mm, deviating from their estimated axes (**Figure 7C**). By the subtraction of 3.5 mm (real adjacent contact distance) mean, the adjacent contact distance error was distributed around 0 mm with a Gaussian-like distribution. Notably, 95% of the contact distance fell in the range of 3.5 ± 1 mm, and 50% of the contact distance fell in the range of 3.5 ± 0.3 mm (**Figure 7D**). These two estimates show comparable results with the Contact Position Estimator (CPE) Module of 3D Slicer (Narizzano et al., 2017).

SEEG Analysis Validation

To evaluate the accuracy of predicting SOZ using EI and HI methods, the selection of the clinician of the SOZ electrode contacts of patients was used as the ground truth. The receiver operator curve (ROC) and the corresponding area under the curve (AUC) were further used to evaluate the consistency between the index-based prediction and the clinical diagnosis. The average AUC of EI and HI on five patients are 0.83 and 0.80, respectively (with EI of S2 excluded) (**Figures 8A,B**). We could observe that on patient S1, both EI and HI have achieved excellent SOZ prediction results, which suggests a valid estimation of SOZs using both methods. The AUC value of S2 based on EI is close to 0.5 and has no predictive effect, due to the fact that the ictal data of S2 displays similar seizure onset activities within every single channel, and the EI method cannot tell the difference from either their timing orders or energy strengths. In contrast, the AUC of S2 based on interictal HI reaches 0.83, which is highly consistent with the clinically annotated SOZs. The case of S2 suggests that when the ictal data cannot provide sufficient diagnostic information, the interictal data can be used to provide extra information for SOZ location, showing the essential value of the interictal SEEG data analysis. In addition, the AUC value of S3 based on HI is 0.49, while its AUC based on HI reaches 0.99. The HI results of S3 performed poorly because those false-positive channels recorded plentiful high-frequency noises. The cases of S2 and S3 suggested the cross-reference value of EI and HI. Finally, we displayed SOZ predictions on reconstructed cortical volume for clinicians to verify the results with imaging evidence (**Figure 8C**). For the case of S2, we marked the clinically annotated contacts as larger spheres and the HI-based SOZs as red spheres, which shows consistency between these two groups. Moreover, we tried a similar EI module in a software, AnyWave (Colombet et al., 2015), to our ictal dataset, and it shows that the EI predictions of BrainQuake have higher ROCs in most cases (**Figures 9A,B**). The comparisons also show that the AUC of BrainQuake EI and HI both is significantly higher than that of AnyWave EI ($p = 0.0078$ and $p = 0.0391$, respectively, two-sided Wilcoxon signed-rank test, **Figure 9C**).

DISCUSSION AND CONCLUSION

The intracranial SEEG data provide abundant electrophysiological information from the human brain for surgical planning and brain research. With the prevalence of SEEG recording in recent years, a large number of neurodata have been generated while researchers are exploring a way to make the best use of it. The challenge lies in both the fusion of multimodal neurodata and intensive computation during the SEEG analysis. In this study, we have introduced a self-sustained Python toolbox, i.e., BrainQuake, integrating multiple approaches to form a complete solution. For the structural

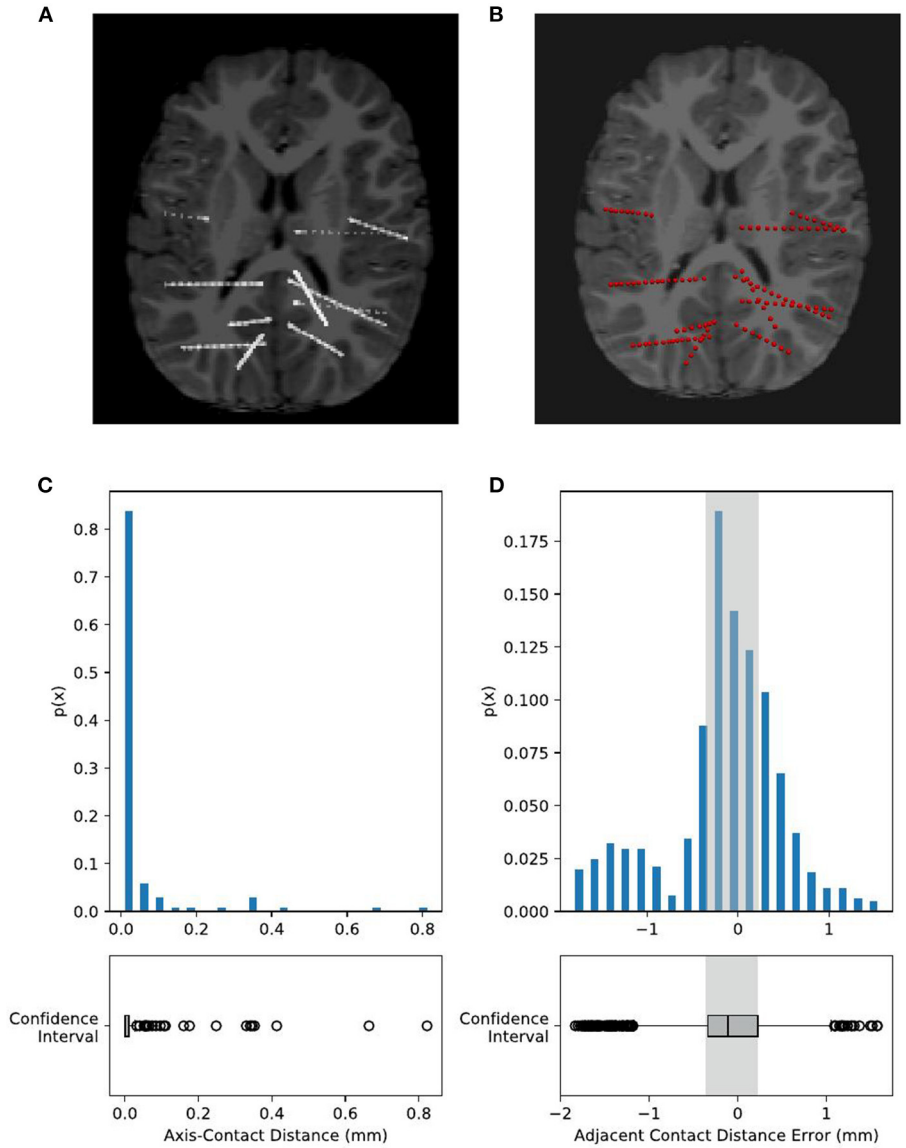


FIGURE 7 | Validation of electrode localization accuracy. Visual checking of the electrodes and contacts of an example subject projected onto the CT image of an individual. The raw CT brain **(A)** shows electrode positions as highlighted line-shaped voxels. Our recognized electrodes (red spheres) are plotted on **(B)**, showing that they are overlapped with each other. Contact positions are quantitatively estimated by two metrics, namely, axis-contact distance and adjacent contact distance error. **(C)** Axis-contact distance estimates the distribution of deviation distance between each contact and its regressed electrode shaft line. Of note, 95% of the contacts were less than 0.1 mm, deviating from their estimated shaft line. **(D)** Adjacent inter-contact distance error estimates the distribution of the distance between each pair of adjacent contacts. The actual adjacent contact distance size, 3.5 mm, is subtracted from the estimated distances, so here we have shown the distribution of the adjacent contact distance error. Notably, 95% of the contact distance fell in the range of 0 ± 1 mm, and 50% of the contact distance fell in the range of 0 ± 0.3 mm, i.e., the adjacent contact distance distribution is 3.5 ± 1 mm (95%) and 3.5 ± 0.3 mm (50%).

data, the electrode module and the surface module provide fast and automated pipelines for surface reconstruction and electrode localization, with only raw MRI T1 and CT images needed for processing. For the functional data, both ictal and interictal modules exploit the long range of SEEG data and provide a presurgical estimation of SOZs. Blending structural and functional results, we provided neurosurgeons with a comprehensive tool for surgical planning. Neuroscientists who are using SEEG to study human brains will also be benefited from our toolbox.

The electrode localization approach implemented in BrainQuake divides the problem into two parts, namely, a global level of electrode clustering and a local level of contact segmentation. BrainQuake innovates in the level of automatic electrode voxel clustering. The semiautonomous methods require either additional input messages or a graphical user

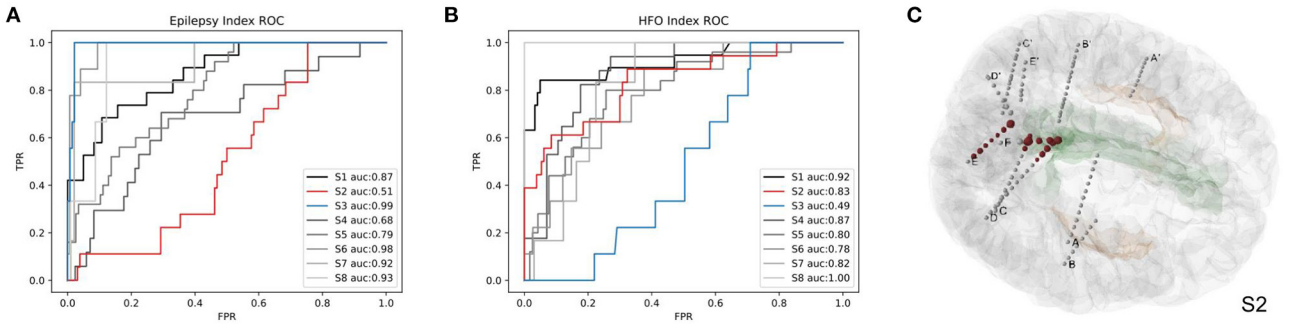


FIGURE 8 | Validations of the SOZ prediction results of BrainQuake comparing with clinically annotated SOZs. **(A)** The receiver operator curve (ROC) and area under the curve (AUC) results of SOZ prediction, based on EI. Case S2 shows a low AUC of 0.51 (low predictive effect), while HI-guided prediction of S2 is 0.83, which is highly consistent with the clinically annotated SOZs. That is because the ictal data of S2 displays similar seizure onset activities within each channel, and the EI method cannot tell the difference from either their timing orders or their energy strengths. The case of S2 suggests that when the ictal data cannot provide sufficient diagnostic information, the interictal data can be used to provide extra information for SOZ location, showing the essential value of interictal SEEG data analysis. **(B)** The ROC and AUC results of SOZ prediction, based on HI. Case S3 shows a low AUC of 0.49 based on HI but a high AUC of 0.99 based on EI. The HI results of S3 performed poorly because those false positive channels recorded plentiful high-frequency noises. EI and HI methods provide prediction results from different perspectives of views, so we recommended surgeons take a comprehensive consideration on both of them. **(C)** The HI results of S2 (marked with larger scales of contacts) and cortical reconstruction are displayed at the same time with clinically annotated SOZs (marked with red color).

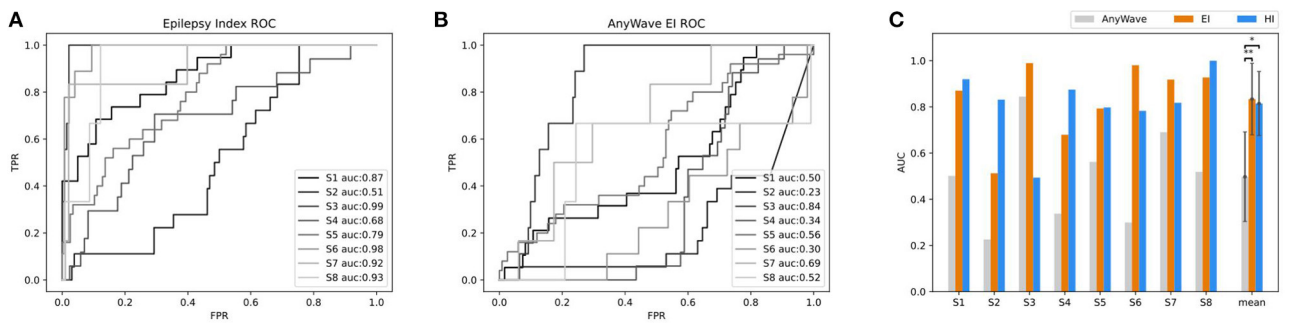


FIGURE 9 | Comparisons between AnyWave EI method and the EI and HI of BrainQuake. **(A)** The ROC and AUC results of SOZ predictions, based on BrainQuake EI. **(B)** The ROC and AUC results of SOZ prediction, based on AnyWave EI. In most cases, BrainQuake EI shows a greater prediction effect than AnyWave EI. **(C)** Comparisons of AUC values between AnyWave EI and BrainQuake EI and HI. A Wilcoxon signed-rank test was performed between the prediction results of AnyWave and BrainQuake. The AUC of BrainQuake EI and HI are both significantly higher than that of AnyWave EI ($p = 0.0078$ and $p = 0.0391$, respectively, two-sided Wilcoxon signed-rank test, $^{*}p < 0.05$, $^{**}p < 0.01$).

interface (GUI) to complete this process, i.e., the efficiency and user experience of which highly depends on the quality of images and preprocessing steps. Our algorithm, which is the combination of 3D Hough Transform and Gaussian Mixture Model, managed to take advantage of both geometric prior and graphical information embedded in CT images. The Hough Transform helps to detect the geometric characteristic of the objects in the image. Whatever the image resolution is high or low, electrode shafts are always straight and highlighted from the background. From this perspective, a pattern recognition algorithm can, in fact, be used to exploit the image instead of scanning it slice by slice. To our knowledge, this valid and useful geometric property has never been utilized in any other SEEG electrode localization method before. The Hough Transform makes electrode shafts be recognized automatically, although it may not return us a precise result. The recognized directions may deviate slightly from the shaft, or a recognized centroid may not be in the exact center of the actual electrode. However, the result can be much close to the true state, which is a good starting point for initializing the clustering algorithm. Thus, we removed the complicated manual intervention, that is, to replace the procedure of telling a software where the electrodes locate with automatic splicing of algorithms, and the pipeline consumes much less time than previous tools.

As for the subsequent step of contact segmentation for every single electrode, the algorithm of the center-of-mass convergence (Arnulfo et al., 2015) has shown interpretable principles and valid results. In our pipeline, we applied this algorithm to each electrode one by one after electrode clustering and acquire the final contact coordinates. We used axis-contact distance and adjacent contact distance error to estimate the geometric characteristics of the segmentation results. However, those two parameters are, in fact, the indirect ways of validating whether the contacts are properly located. Several factors may influence the

error distributions. An electrode can bend slightly in the brain, in which case there is a possibility that fluctuations occur in the distributions of both parameters. It can generate some outliers in the distribution of axis-contact distance since the contacts are no longer scattered along a straight line and the deviations of contacts from the regressed line, in fact, exist. Moreover, due to the bending, the adjacent contact distance may shrink slightly as the contacts bear the force to be compressed to each other. Reflecting on **Figure 7D**, there are more distance errors lying in the negative half range than in the positive half range. In other cases, failures do exist due to the quality of the raw CT images. There are possibilities that the algorithm cannot find a local center-of-mass in a region and keep looking for highlights along the direction and finally converge to the next contact. This can explain the positive outliers in **Figure 7D**. We encountered a worse situation that the two regions of highlights were too close to each other and so the converging point just kept jumping from one optimal to another. We fixed this problem by implementing a counting index of convergence in the algorithm setting a forcing scheme to stop the infinite loop and choosing a voxel with higher voxel values just in case. We could notice that the design of the center-of-mass convergence algorithm does have its deficits and may not give us highly precise results. The recommended redeeming method is still visual checking. As for the essentiality of precise contact locations and then the locations of potential SOZs, one must not skip the procedure of manual checking. By projecting the contact results onto the registered CT image on a NIfTI image reading software such as "Freeview" (**Figure 7B**), we could go through the slices to check if the contacts recognized are matched with the highlighted voxels in the image. If an error is detected, surely one can erase a misplaced contact and add a new one by hand.

The automatic SOZ prediction methods usually use the onset order of high-frequency activity at each contact during the seizure or the specific distribution of abnormal activity during the interictal period as pathological features (Bartolomei et al., 2008; Barkmeier et al., 2012; Navarrete et al., 2016) These methods have already been integrated into some software independently (Tadel et al., 2011; Colombet et al., 2015). We tried a similar EI module in a software, AnyWave (Colombet et al., 2015), to our ictal dataset, and the comparison results show that the EI predictions of BrainQuake have higher ROCs in most of the cases (**Figures 9A,B**). Although the seizure data are considered to be more relevant to SOZ prediction, it may be difficult to capture or it may not provide enough information for the diagnosis, resulting in a relatively low AUC. Meanwhile, a large amount of interictal SEEG has not been fully utilized. The pathological information extracted from the long-term data may also have good predictive power on SOZ and is more immune to noises than the ictal data. As shown in our results (**Figure 9C**), HI derived from the interictal data is a good supplement to the EI method, and clinicians can compare the consistency between them. BrainQuake may serve as a platform for exploring the causal relationships between these two kinds of predictions and ultimately lead to better clinical diagnoses.

The processing of the long-term interictal data also gives rise to the challenge of computing power. The progress in deep learning has led to the development of high-performance parallel computing, and meanwhile, the acceleration capability of GPUs may be a solution to massive SEEG data and its high-load computing. At present, the mechanisms of seizures and interictal discharges are still unclear, and they may reflect different aspects of the epileptic network (Jiruska et al., 2017; Grinenko et al., 2018). In the future, we plan to implement a GPU module for the long-term interictal SEEG analysis in BrainQuake, and the prediction methods from the perspective of epileptic networks are to be explored.

BrainQuake is designed to be an auxiliary tool for epilepsy neurosurgeons and technicians, trying to convey a presurgical evaluation solution with blended functional and structural neurodata. Most current software or toolboxes focus on one or a few steps, developing splendid algorithms or techniques for data processing, but in clinical practice, it is a cumbersome task to merge all kinds of results into one system or coordinate. Also, several steps consume a lot of time and effort to do repeated work, resulting in an inefficient working procedure. BrainQuake commits to freeing surgeons and technicians from tedious and time-consuming work, allowing them to concentrate on the steps that rely more on common sense and medical expertise short in machine algorithms. In the upcoming era of big neurodata, this kind of human-computer synergy is an efficient approach to data utilization, and we believe that it will eventually promote the fields of both neurology and neuroscience.

DATA AVAILABILITY STATEMENT

The example dataset presented in this study can be found in https://doi.org/10.5281/zenodo.5675459. The codes can be found on Github (https://github.com/HongLabTHU/Brainquake).

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Ethics Committee of Tsinghua Yuquan Hospital. Written informed consent to participate in this study was provided by the participants' legal guardian/next of kin.

AUTHOR CONTRIBUTIONS

BH, FC, and KW conceived the work, contributed to drafting, and revising the article. FC and KW designed the software. FC developed the surface module and the electrode module. KW and TZ developed the ictal module. KW developed the interictal module. HW and WZ collected the experimental data. All authors contributed to the article and approved the submitted version.

FUNDING

This study was financially supported by the National Key R&D Program of China (2017YFA0205904 to BH).

ACKNOWLEDGMENTS

We thank Yanqin Lei, Zhao Wang, Bingqing Zhang, Jianjun Bai, Siyu Wang, Yiou Liu, and Jie Shi for their assistance and suggestions in data analysis and software designing, as well as Yuxiang Yan and Wenzheng Li for the comments on the manuscript. We also thank the reviewers/editors for the critical reading of the manuscript. We appreciate the time and dedication of the patients and staff at the Epilepsy Center, Yuquan Hospital, Tsinghua University, Beijing, China.

REFERENCES

Akkol, S., Kucyi, A., Hu, W., Zhao, B., Zhang, C., Sava-Segal, C., et al. (2021). Intracranial electroencephalography reveals selective responses to cognitive stimuli in the periventricular heterotopias. *J. Neurosci.* 41, 3870–3878. doi: 10.1523/JNEUROSCI.2785-20.2021

Arnulfo, G., Narizzano, M., Cardinale, F., Fato, M. M., and Palva, J. M. (2015). Automatic segmentation of deep intracerebral electrodes in computed tomography scans. *BMC Bioinformat.* 16, 1–12. doi: 10.1186/s12859-015-0511-6

Bancaud, J., Talairach, J., Bonis, A., Schaub, C., Szikla, G., Morel, P., et al. (1965). *La stéreoencephalographie dans l'épilepsie*. Paris: Mattson, 113–146.

Barkmeier, D. T., Shah, A. K., Flanagan, D., Atkinson, M. D., Agarwal, R., Fuerst, D. R., et al. (2012). High inter-reviewer variability of spike detection on intracranial eeg addressed by an automated multi-channel algorithm. *Clin. Neurophysiol.* 123, 1088–1095. doi: 10.1016/j.clinph.2011.09.023

Bartolomei, F., Chauvel, P., and Wendling, F. (2008). Epileptogenicity of brain structures in human temporal lobe epilepsy: a quantified study from intracerebral eeg. *Brain* 131, 1818–1830. doi: 10.1093/brain/awn111

Behrens, E., Zentner, J., Van Roost, D., Hufnagel, A., Elger, C. E., and Schramm, J. (1994). Subdural and depth electrodes in the presurgical evaluation of epilepsy. *Acta Neurochirurg.* 128, 84–87. doi: 10.1007/BF01400656

Blenkmann, A. O., Phillips, H. N., Princich, J. P., Rowe, J. B., Bekinschtein, T. A., Muravchik, C. H., et al. (2017). ielectrodes: a comprehensive open-source toolbox for depth and subdural grid electrode localization. *Front. Neuroinformat.* 11:14. doi: 10.3389/fninf.2017.00014

Cai, Z., Sohrabpour, A., Jiang, H., Ye, S., Joseph, B., Brinkmann, B. H., et al. (2021). Noninvasive high-frequency oscillations riding spikes delineates epileptogenic sources. *Proc. Natl. Acad. Sci. U. S. A.* 2021:118. doi: 10.1073/pnas.2011130118

Colombet, B., Woodman, M., Badier, J., and Bénar, C. (2015). Anywave: a cross-platform and modular software for visualizing and processing electrophysiological signals. *J. Neurosci. Method.* 242, 118–126. doi: 10.1016/j.jneumeth.2015.01.017

Cossu, M., Fuschillo, D., Casaceli, G., Pelliccia, V., Castana, L., Mai, R., et al. (2015). Stereoelectroencephalography-guided radiofrequency thermocoagulation in the epileptogenic zone: a retrospective study on 89 cases. *J. Neurosurg.* 123, 1358–1367. doi: 10.3171/2014.12.JNS141968

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

Dalitz, C., Schramke, T., and Jeltsch, M. (2017). Iterative hough transform for line detection in 3d point clouds. *Image Proces.* 7, 184–196. doi: 10.5201/ipol.2017.208

Darcey, T. M., and Roberts, D. W. (2010). Technique for the localization of intracranially implanted electrodes. *J. Neurosurg.* 113, 1182–1185. doi: 10.3171/2009.12.JNS091678

Dykstra, A. R., Chan, A. M., Quinn, B. T., Zepeda, R., Keller, C. J., Cormier, J., et al. (2012). Individualized localization and cortical surface-based registration of intracranial electrodes. *Neuroimage* 59, 3563–3570. doi: 10.1016/j.neuroimage.2011.11.046

Fischl, B. (2012). Freesurfer. *Neuroimage* 62, 774–781. doi: 10.1016/j.neuroimage.2012.01.021

Grinenko, O., Li, J., Mosher, J. C., Wang, I. Z., Bulacio, J. C., Gonzalez-Martinez, J., et al. (2018). A fingerprint of the epileptogenic zone in human epilepsies. *Brain* 141, 117–131. doi: 10.1093/brain/awx306

Hamilton, L. S., Chang, D. L., Lee, M. B., and Chang, E. F. (2017). Semi-automated anatomical labeling and inter-subject warping of high-density intracranial recording electrodes in electrocorticography. *Front. Neuroinformat.* 11:62. doi: 10.3389/fninf.2017.00062

Henschel, L., Conjeti, S., Estrada, S., Diers, K., Fischl, B., and Reuter, M. (2020). Fastsurfer-a fast and accurate deep learning based neuroimaging pipeline. *NeuroImage* 219:117012. doi: 10.1016/j.neuroimage.2020.117012

Illingworth, J., and Kittler, J. (1988). A survey of the hough transform. *Comput. Vis. Graph. Image Proces.* 44, 87–116. doi: 10.1016/S0734-189X(88)80033-1

Jeltsch, M., Dalitz, C., and Pohle-Fröhlich, R. (2016). "Hough parameter space regularisation for line detection in 3d," in *VISIGRAPP (4: VISAPP)* (Rome: SciTePress), 345–352. doi: 10.5220/0005679003450352

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

Jiruska, P., Alvarado-Rojas, C., Schevon, C. A., Staba, R., Stacey, W., Wendling, F., et al. (2017). Update on the mechanisms and roles of high-frequency oscillations in seizures and epileptic disorders. *Epilepsia* 58, 1330–1339. doi: 10.1111/epi.13830

Kwan, P., and Brodie, M. J. (2000). Early identification of refractory epilepsy. *N. Engl. J. Med.* 342, 314–319. doi: 10.1056/NEJM200002033420503

Li, G., Jiang, S., Chen, C., Brunner, P., Wu, Z., Schalk, G., et al. (2019). ieegview: an open-source multifunction gui-based matlab toolbox for localization and visualization of human intracranial electrodes. *J. Neural Eng.* 17:016016. doi: 10.1088/1741-2552/ab51a5

Narizzano, M., Arnulfo, G., Ricci, S., Toselli, B., Tisdall, M., Canessa, A., et al. (2017). Seeg assistant: a 3dslicer extension to support epilepsy surgery. *BMC Bioinformat.* 18, 1–13. doi: 10.1186/s12859-017-1545-8

Navarrete, M., Alvarado-Rojas, C., Le Van Quyen, M., and Valderrama, M. (2016). Ripplelab: a comprehensive application for the detection, analysis and classification of high frequency oscillations in electroencephalographic signals. *PLoS ONE* 11:e0158276. doi: 10.1371/journal.pone.0158276

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Machine Learn. Res.* 12, 2825–2830.

Qin, C., Tan, Z., Pan, Y., Li, Y., Wang, L., Ren, L., et al. (2017). Automatic and precise localization and cortical labeling of subdural and depth intracranial electrodes. *Front. Neuroinformat.* 11:10. doi: 10.3389/fninf.2017.00010

Remakanthakurup Sindhu, K., Staba, R., and Lopour, B. A. (2020). Trends in the use of automated algorithms for the detection of high-frequency oscillations associated with human epilepsy. *Epilepsia* 61, 1553–1569. doi: 10.1111/epi.16622

Reynolds, D. A. (2009). Gaussian mixture models. *Encycl. Biometr.* 741, 659–663. doi: 10.1007/978-0-387-73003-5_196

Roehri, N., and Bartolomei, F. (2019). Are high-frequency oscillations better biomarkers of the epileptogenic zone than spikes? *Curr. Opin. Neurol.* 32, 213–219. doi: 10.1097/WCO.0000000000000663

Roehri, N., Pizzo, F., Bartolomei, F., Wendling, F., and Bénar, C.-G. (2017). What are the assets and weaknesses of hfo detectors? a benchmark framework based on realistic simulations. *PLoS ONE* 12:e0174702. doi: 10.1371/journal.pone.0174702

Rosenow, F., and Lüders, H. (2001). Presurgical evaluation of epilepsy. *Brain* 124, 1683–1700. doi: 10.1093/brain/124.9.1683

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

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nat. Method.* 17, 261–272. doi: 10.1038/s41592-020-0772-5

Wang, S., So, N. K., Jin, B., Wang, I. Z., Bulacio, J. C., Enatsu, R., et al. (2017). Interictal ripples nested in epileptiform discharge help to identify the epileptogenic zone in neocortical epilepsy. *Clin. Neurophysiol.* 128, 945–951. doi: 10.1016/j.clinph.2017.03.033

Wang, S., Wang, I. Z., Bulacio, J. C., Mosher, J. C., Gonzalez-Martinez, J., Alexopoulos, A. V., et al. (2013). Ripple classification helps to localize the seizure-onset zone in neocortical epilepsy. *Epilepsia* 54, 370–376. doi: 10.1111/j.1528-1167.2012.03721.x

Wang, S., Zhao, M., Li, T., Zhang, C., Zhou, J., Wang, M., et al. (2020). Stereotactic radiofrequency thermocoagulation and resective surgery for patients with hypothalamic hamartoma. *J. Neurosurg.* 134, 1019–1026. doi: 10.3171/2020.2.JNS193423

Zhang, Y., Zhou, W., Wang, S., Zhou, Q., Wang, H., Zhang, B., et al. (2019). The roles of subdivisions of human insula in emotion perception and auditory processing. *Cerebr. Cortex* 29, 517–528. doi: 10.1093/cercor/bhx334

Zhao, T., Wang, H., Wang, K., Yang, X., Zhou, W., and Hong, B. (2019). "Cross-modal consistency of epileptogenic network in seeg and resting-state fmri," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)* (San Francisco, CA: IEEE), 953–956. doi: 10.1109/NER.2019.8716989

Zöllei, L., Iglesias, J. E., Ou, Y., Grant, P. E., and Fischl, B. (2020). Infant freesurfer: an automated segmentation and surface extraction pipeline for t1-weighted neuroimaging data of infants 0–2 years. *Neuroimage* 218:116946. doi: 10.1016/j.neuroimage.2020.116946

Conflict of Interest: 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.

frontiers in Neuroinformatics

TECHNOLOGY AND CODE
published: 13 January 2022
doi: 10.3389/fninf.2021.770608

BIDScoin: A User-Friendly Application to Convert Source Data to Brain Imaging Data Structure

Marcel Peter Zwiers[1], Stefano Moia[2] and Robert Oostenveld[1,3]*

[1] Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, Netherlands, [2] Basque Center on Cognition, Brain and Language, San Sebastian, Spain, [3] NatMEG, Karolinska Institutet, Stockholm, Sweden

Analyses of brain function and anatomy using shared neuroimaging data is an important development, and have acquired the potential to be scaled up with the specification of a new Brain Imaging Data Structure (BIDS) standard. To date, a variety of software tools help researchers in converting their source data to BIDS but often require programming skills or are tailored to specific institutes, data sets, or data formats. In this paper, we introduce BIDScoin, a cross-platform, flexible, and user-friendly converter that provides a graphical user interface (GUI) to help users finding their way in BIDS standard. BIDScoin does not require programming skills to be set up and used and supports plugins to extend their functionality. In this paper, we show its design and demonstrate how it can be applied to a downloadable tutorial data set. BIDScoin is distributed as free and open-source software to foster the community-driven effort to promote and facilitate the use of BIDS standard.

Keywords: BIDS, GUI, conversion, neuroimaging, data sharing, open-source software, Python, plugin

OPEN ACCESS

Edited by:
William T. Katz,
Janelia Research Campus,
United States

Reviewed by:
Christopher Markiewicz,
Stanford University, United States
Yaroslav O. Halchenko,
Dartmouth College, United States

****Correspondence:***
Marcel Peter Zwiers
marcel.zwiers@donders.ru.nl

Received: *04 September 2021*
Accepted: *15 November 2021*
Published: *13 January 2022*

Citation:
Zwiers MP, Moia S and Oostenveld R (2022) BIDScoin: A User-Friendly Application to Convert Source Data to Brain Imaging Data Structure.
Front. Neuroinform. 15:770608.
doi: 10.3389/fninf.2021.770608

INTRODUCTION

In the last few decades, neuroimaging data have become an increasingly rich source of information for studying the working of the brain in health and disease. Typically, the acquisition of such data sets is expensive and often difficult to collect from a large number of participants. Contemporary neuroscientific and clinical research questions are based on ever advancing analysis methods that require the availability of data sets that are very large (also known as "big data") or of superior quality, or both. Several initiatives have been undertaken to address this problem by pooling the data from individual studies across the globe and by sharing data in online repositories (see, e.g., Turner et al., 2016, for a special issue overview).

The initial lack of data-structure standardization, data-sharing tools, and data-sharing mindset (Poline et al., 2012; Nichols et al., 2017; White et al., 2020) have led to the use of a large variety of file formats and data management methods, and to the lack of metadata descriptions, leaving researchers with the daunting task of adapting all the collected data in a custom format to run their analysis pipelines. Recently, the Brain Imaging Data Structure (BIDS; Gorgolewski et al., 2016) was introduced to alleviate this task to increase data sharing and usage and to facilitate reproducibility studies.

In essence, BIDS is a specification that prescribes how data collections should be organized and formatted on disk, in a computer and human readable way: it specifies the folder structure, the file names, the metadata fields, and its file formats. The BIDS standard was initially developed for MRI data but has since been embraced by the wider neuroimaging community, as indicated by

extensions of the standard to MEG, EEG, iEEG, genetic, and PET data (Niso et al., 2018; Holdgraf et al., 2019; Pernet et al., 2019; Knudsen et al., 2020; Moreau et al., 2020), by a large number of BIDS Extension Proposals,[1] a surge in BIDS apps (Gorgolewski et al., 2017), and a wide adoption of BIDS being used in publications. BIDS standard has moved the burden of homogenizing the data from the end user to the researchers who have collected the data set – and, importantly, who have the best knowledge about that data. In addition, the development of BIDS apps[2] has provided researchers with easy to use, standardized processing pipelines that are typically well tested and documented.

Nevertheless, a limiting factor in the adoption of BIDS standard is that many of the neuroscientists who collect data do not have the programming skills to reformat their data in an efficient or automated manner. Various BIDS conversion command-line tools[3] to support researchers have been made available, ranging from institute- or study-specific solutions, to community developed software, and from poorly documented tools to more advanced packages with programmatic interfaces. Among these, many use the well-known dcm2niix converter (Li et al., 2016) under the hood to perform the actual data conversion, such as the popular HeuDiConv (Halchenko et al., 2020), dcm2bids,[4] and the related bidskit[5] tools. HeuDiConv is a powerful tool but requires Python programming skills, albeit basic, and its rule-base heuristics design has a relatively steep learning curve and requires technical knowledge about the data. Dcm2bids uses a mapping approach that is easier to use although users still need to manually write their own configuration files. Solutions also exist for non-MRI data such as EEG or MEG data converters such as MNE-BIDS (Appelhoff et al., 2019), FieldTrip,[6] and EEGLAB[7].

A common limitation of the available tools is that they generally lack graphical user interfaces (GUIs) that can lower the barrier to adopt BIDS standard. To our knowledge, only a few converters come with a GUI. A service named as ezBIDS[8] allows researchers to use a web browser to upload their DICOM data to a web server, to configure the dcm2niix-based data conversion, and to download a converted BIDS data set. Furthermore, a plugin[9] for the Horos/OsiriX DICOM viewer uses dcm2niix to convert DICOM data to BIDS. Finally, pyBIDSconv,[10] is an MRI-centered wrapper around dcm2niix, DataLad-hirni[11] is an extension for DataLad (Halchenko et al., 2021), and Biscuit[12] is an MEG-centered wrapper around MNE-BIDS. However, these three converters no longer seem to be under active development and do not support recent extensions to BIDS standard.

With the BIDScoin application suite presented in this paper, we aim to further promote the usage of BIDS by providing a flexible framework to convert any kind of source data to BIDS in a user-friendly way, which requires no previous programming knowledge. To achieve this goal, BIDScoin uses an intelligent mapping approach to associate raw source data types with BIDS target data types. The approach exploits as much of the digitally available information about the data as possible as well as the information that is typically known only by the researcher. The mapping approach of BIDScoin is intuitive for neuroimaging researchers as (1) it resembles the way they often think about their data types (they can recognize the data types they have collected when they see them, but do not know how to uniquely and reliably identify them technically), (2) it is simple and flexible as a virtually unlimited number of concurrent mappings can be established, and (3) it offers a GUI for users to directly and easily edit the mappings to their needs.

METHOD

All BIDScoin codes are freely available at github[13] and pypi,[14] and the documentation can be found on Read the Docs.[15] The latest BIDScoin version 3.7 as described in this paper is written in Python 3.6 and dependent on the freely available PyQt5 (Riverbank Computing Limited, Dorchester, England) software library for the GUI.

The BIDScoin Workflow

The workflow of BIDScoin to convert source data into BIDS standard consists of three steps (**Figure 1**):

(1a) To start with, the researcher runs a command-line application named as "bidsmapper" to perform the data discovery on their source data set (i.e., the folder containing all the input files). In this step, a so-called "template bidsmap" is used to scan the entire source data set and automatically create what will be referred to as a "study bidsmap." Conceptually, the template bidsmap can be thought of as a set of broad filters, each of which maps a source data type onto a single BIDS output data type (e.g., anat, func, fmap, and dwi), onto an "exclude" data type that is not converted to BIDS, or, if none of the filters match, onto an unknown "extra_data" data type. Whenever a template filter matches with a source data type, the bidsmapper narrows the filter to exactly match to this particular source data type only and adds it to the study bidsmap if not present there yet. In this way, a mapping shortlist is built up in the study bidsmap, representing all of the unique source data types that are present in the source data folder. Note that a single broad filter from the template bidsmap can result in multiple narrow filters in the study

[1] https://bids.neuroimaging.io/get_involved.html#extending-the-bids-specification.

[2] https://bids-apps.neuroimaging.io.

[3] https://bids.neuroimaging.io/benefits.html#converters.

[4] https://github.com/cbedetti/Dcm2Bids.

[5] https://github.com/jmtyszka/bidskit.

[6] https://www.fieldtriptoolbox.org/reference/data2bids.

[7] https://github.com/arnodelorme/bids-matlab-tools.

[8] https://github.com/brainlife/ezbids.

[9] https://github.com/mslw/horos-bids-output.

[10] https://github.com/DrMichaelLindner/pyBIDSconv.

[11] https://github.com/psychoinformatics-de/datalad-hirni.

[12] https://github.com/Macquarie-MEG-Research/Biscuit.

[13] https://github.com/Donders-Institute/bidscoin.

[14] https://pypi.org/project/bidscoin.

[15] https://bidscoin.readthedocs.io.

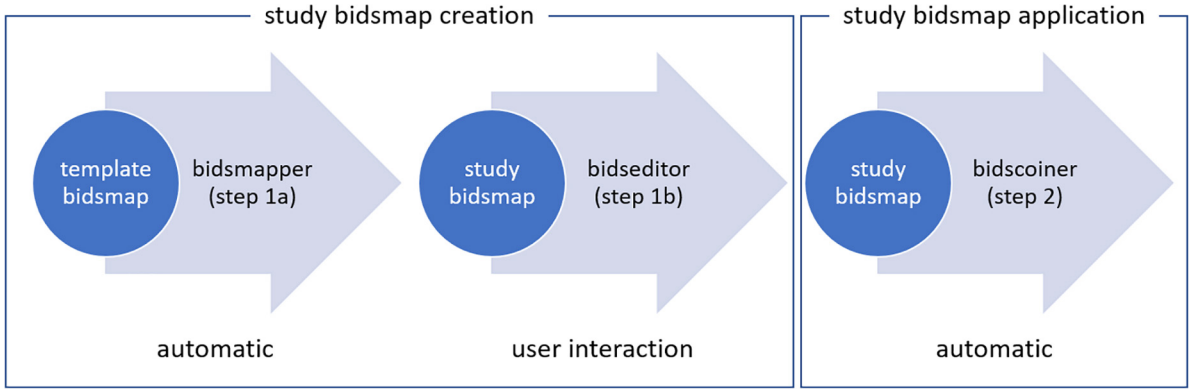


FIGURE 1 | Creation and application of a study bidsmap. The user runs the "bidsmapper" executable with a "template bidsmap" as an input and with a "study bidsmap" as an output with the suggested Brain Imaging Data Structure (BIDS) data types, entities, and metadata. The study bidsmap is verified and edited interactively with the "bidseditor" graphical user interface (GUI). Finally, the study bidsmap is passed to the "bidscoiner" to convert the source data to BIDS.

bidsmap, for instance when two similar anatomical MRI scans are collected with different spatial resolutions. In the rest of this paper, we will refer to a bidsmap filter that maps to a BIDS data type (including the output file names and metadata) as a "BIDS mapping." A template bidsmap is generic and typically created once, whereas a study bidsmap is tailored to the data at hand and therefore stored together with the output data.

(1b) After the study bidsmap has been created, a GUI application named as "bidseditor" is launched, either automatically by the bidsmapper or manually. The bidseditor reads in the study bidsmap and opens a main window that shows the shortlist of the discovered source data types and their suggested mappings to BIDS (**Figure 2**). From the main window, researchers can open subwindows to enrich or correct each of the suggested mappings using the knowledge they have about the data (**Figure 3**). In BIDScoin, prior (e.g., research center-specific) knowledge about the data can be represented in the template bidsmap: the more of this knowledge is represented, the larger the number of correctly suggested mappings will be, and the lesser edits the researcher needs to make. When the template bidsmap is unsuited or lacks any prior intelligence, all source data types will be classified as "extra_data" and the researcher will have to edit each mapping to the correct BIDS data type. Still, in such a worst-case scenario, the researcher has to perform only a limited amount of work on a short list of items. The bidsmap and all the user edits are immediately validated against the public BIDS schema files to ensure the specification of all mandatory fields and produce the correct metadata and valid naming of all the output files.

(2) After the data discovery and editing are done, the final step in the workflow is to call the "bidscoiner" application to automatically convert ("coin") the source data set to a BIDS data set, as specified by the mappings in the study bidsmap. Note that as the number of mappings is independent from the number of subjects or sessions, the bidscoiner can be re-run every time new subects or sessions are added to the source dataset, without the need to re-run the bidsmapper or editor. If new scan protocols are employed for subsequent data, the researcher can repeat steps 1a and 1b first, which will reload the previously edited mappings and add the mappings for the new data samples to the list.

The Bidsmap

Brain Imaging Data Structure Mapping

Thus far, we have referred to the bidsmap as a collection of BIDS mappings (filters) that define how the different source data types should be converted to the BIDS output data. Source data typically comes with two sources of information about the datatype and acquisition parameters, namely (1) metadata that is inherent to the filesystem, such as parts of the folder or file name, and (2) metadata that is intrinsic to the data itself, such as information represented in the header of the binary file. Depending on the imaging modality and on the data management plan, researchers often use either one of these sources or both. For instance, as opposed to MRI data in the DICOM format, most EEG data formats contain rather limited header information. Similarly, in the BIDS format, metadata is also stored (3) in the file path and file name and (4) in an accompanying json sidecar file. Hence, to be as versatile as possible, all four sources of information are represented in a BIDS mapping:

1. The file system metadata is contained in an input dictionary named "properties." This dictionary contains file system properties of the data sample, i.e., the file path (in POSIX-style), the file name, the file size on disk, and the number of files in the containing folder. Depending on ones' data management, this information allows or can help to identify the different data types in the source data repository.
2. The intrinsic metadata is contained in an input dictionary named "attributes." This dictionary normally consists of a

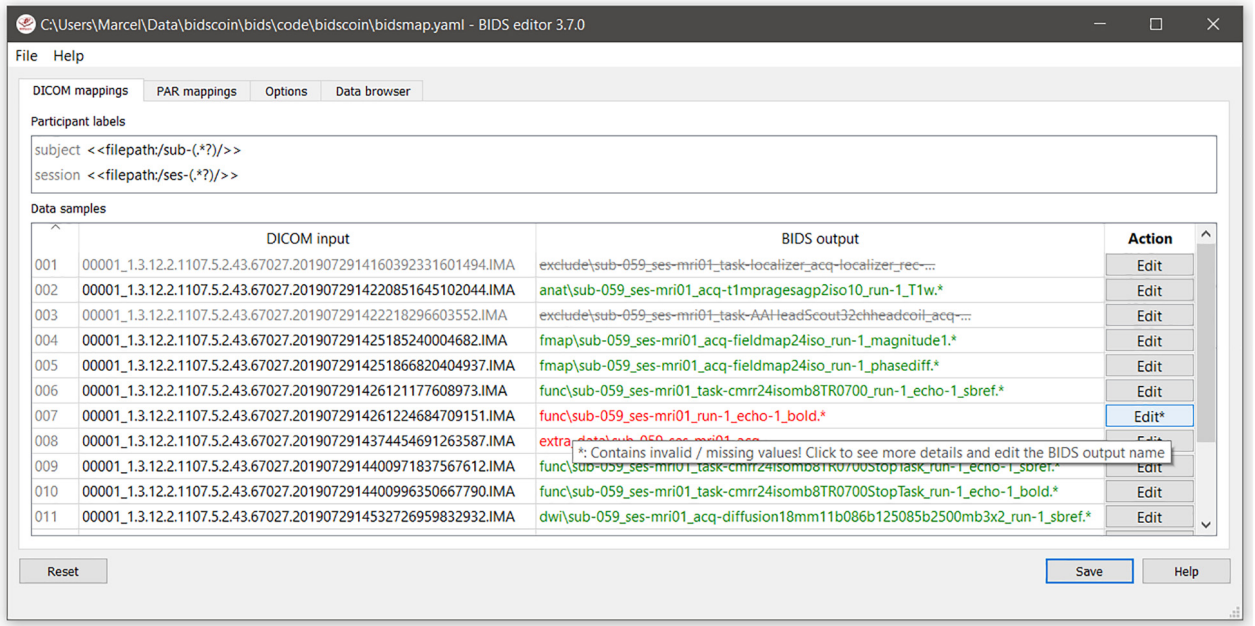


FIGURE 2 | The bidseditor main window with an overview of the data types in the source data (left column) with a preview of the BIDS output names (right column). The green or red color indicates whether manual editing of the BIDS mapping is necessary, while the strikeout text indicates that the data type will not be converted, which is useful for handling irrelevant data. The user can edit the "subject" and "session" property values if needed ("session" can be left empty to be omitted) and the result is immediately reflected in the preview. Different tabs represent different data formats in the source data set, i.e., DICOM and PAR, which are represented as separate sections in the bidsmap. In addition, there is a tab to edit the study-specific "Options" and a tab in which the user can browse the organization of the source data and inspection of the data.

minimal subset of the available intrinsic metadata that is effective to identify the different data types in the source data repository.

3. The BIDS entities that define the file name after the conversion are contained in an output dictionary named "bids."
4. The BIDS metadata is contained in an output dictionary named "meta." The meta dictionary contains the custom key-value pairs that are added to a new or an existing json sidecar file by the bidscoiner plugins (further described later).

When source data is scanned by a BIDScoin routine, the keys of these input dictionaries indicate which metadata is to be extracted from the source data and matched against the dictionary value. In this identification procedure, the input dictionary values are interpreted by BIDScoin as regular expression patterns,[16] and as such define the abovementioned broadness of the template or study bidsmap filters.

For instance, in a template bidsmap, a key-value pair of an attribute dictionary could be `{ProtocolName: .*(mprage|T1w).*}`,[17] which would extract the attribute string for "ProtocolName" from the DICOM[18] header and tests if that string contains either a "mprage" or a "T1w" substring. BIDScoin will test all the key-value pairs of the input dictionaries and will consider it an overall match only if all of them tested positively. During the bidsmapper runtime, the existing attribute values are then replaced (expanded) by the full string values that were extracted from the header, e.g., `{ProtocolName: t1_mprage_sag_p2_iso_1.0}`, and then stored in the study bidsmap as a new BIDS mapping.

The expanded values will not contain broadly matching wildcards (.*) or Boolean "OR" operators (|), and hence act as very narrow regular expressions that make exact matches only. Note that the initial pattern from the template contains the prior knowledge that the data type is most likely "T1w" if the DICOM ProtocolName contains a "mprage" or "T1w" substring, but that the exact "t1_mprage_sag_p2_iso_1.0" substring is study-specific and cannot be predicted *a priori*. In this way, BIDScoin will collect all the source data types and will notify any unintended deviation from the data acquisition protocol.

Within a bidsmap, BIDS mappings are hierarchically grouped in BIDS data types, such as "anat," "dwi," and "func," and accompanied with a "subject" and a "session" key-value pair for extracting BIDS subject and session labels. A snippet of a study bidsmap in the YAML[19] format can be seen in **Figure 4**.

[16]https://docs.python.org/3/library/re.html.

[17]The Python syntax that is used throughout this paper of a key-value pair is `{key: value}`. Here, the `.*(mprage|T1w).*` value is taken as a regular expression pattern. In regular expressions, the ".*" in the pattern denotes a wildcard, the "| " denotes an OR statement, and the parenthesis indicate a grouping. The expression is evaluated in Python as: `match = re.fullmatch(pattern, string)`.

[18]http://dicom.nema.org.

[19]http://yaml.org.

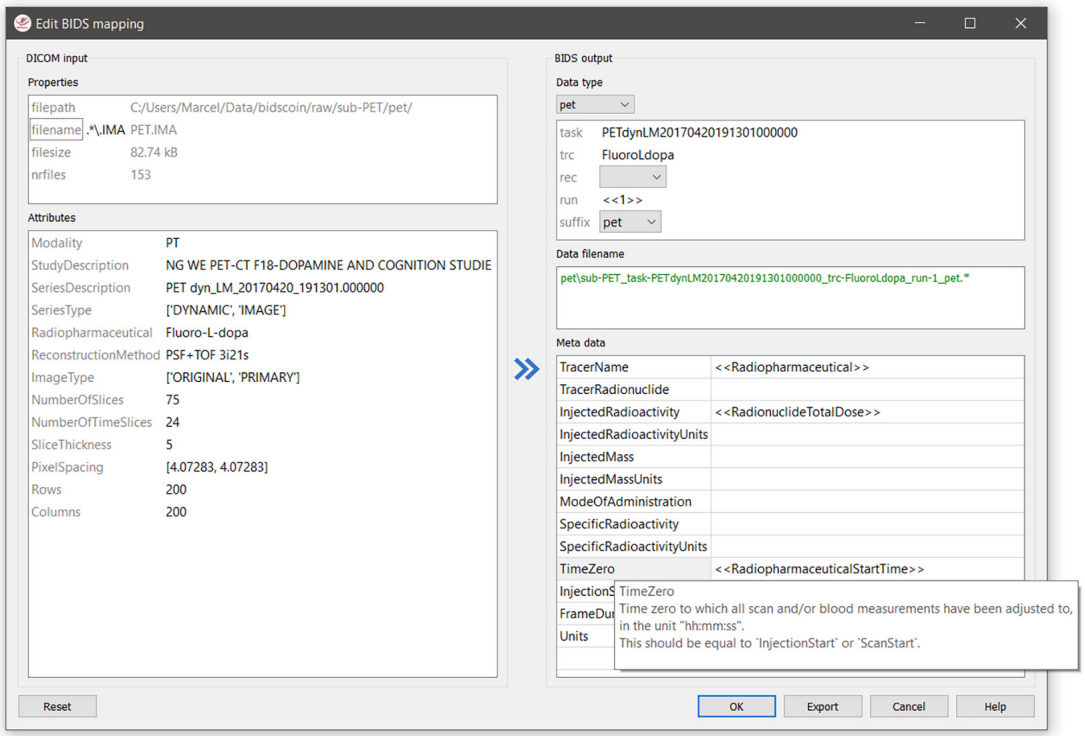


FIGURE 3 | The BIDS mapping edit window featuring filename matching (.*\.IMA) and dynamic metadata values (e.g., "TimeZero"). The BIDS values that are restricted to a limited set are presented with a drop-down menu (here the "Data type," the "rec," and the "suffix" value). The user can immediately see the results of their edits in the preview of the BIDS output file name. A green file name indicates that the name is compliant with BIDS standard, whereas a red name indicates that the user still needs to fill out one or more compulsory BIDS values (with a pop-up window appearing if the user ignores it). Hoovering with the mouse over dictionary keys pops up explanatory text from the BIDS schema files, as highlighted for "TimeZero". Double clicking on the DICOM file name opens a new window displaying the full header information with all attributes. The user can export the customized mapping to a different bidsmap on disk.

Dynamic Values

In the BIDScoin workflow, users can directly set the bidsmap values as they like, but often these values are already available as file attributes or properties or may vary between acquisitions. BIDScoin allows researchers to capture such cases with so-called "dynamic values." Bidsmap values are treated as "dynamic" when they are captured between single (<>) or double brackets (<<>>), in which case the value should correspond to an attribute or a property key for which the value can be extracted from the data. Single brackets will always be extracted directly by BIDScoin routines and are typically part of a template bidsmap. Hence, when the template bidsmap is converted to a study bidsmap, the dynamic values are extracted and presented to the user for further editing. For instance, `{acq: <ProtocolName>}` in the bids output dictionary of the template bidsmap will appear as `{acq: t1mpragesagp2iso10}` in the study bidsmap (**Figure 2**). Double bracket dynamic values will remain as they are and will only be extracted during (bidscoiner) runtime, as explained further below.

Single bracket dynamic values are most useful as an intelligent first guess for the output dictionary values that vary only between data types, but not between acquisitions, such as the MRI sequence parameters "ProtocolName" or "FlipAngle." Double bracket values can be useful for the dictionary values that vary more often, such as between subjects, sessions, or runs[20] of the same data type. For instance, the value `{Comments: <<PatientComments>>}` in the meta dictionary (**Figure 2**) will extract the comments for that specific subject or session, while the value `{subject: <<filepath:/sub-(.*?)/>>}`[21] will extract "003" (i.e., the shortest string between "/sub-" and "/") if the data for that subject is in "/data/raw/sub-003/ses-01." The latter example illustrates how a colon-separated regular expression can be appended to the "filepath" or "filename" property keys to extract a substring as researchers often encode multiple values or key-value pairs in a single filepath or filename. This mechanism to extract substrings is not limited to the file path or file name property keys but can be

[20] In BIDS, a run is defined as: an uninterrupted repetition of data acquisition that has the same acquisition parameters and task (however, events can change from run to run due to a different subject response or randomized nature of the stimuli). Run is a synonym of a data acquisition. Note that "uninterrupted" may look different by modality due to the nature of the recording. For example, in MRI or MEG, if a subject leaves the scanner, the acquisition must be restarted. For some types of PET acquisitions, a subject may leave and re-enter the scanner without interrupting the scan.

[21] In regular expressions, the ".*?" pattern denotes a non-greedy wildcard. The substring that matches with the pattern between the parentheses is captured as the expression is evaluated in Python as: `substring = re.findall(pattern, string)`.

```
DICOM:
# ------------------------------------------------------------------------------
# DICOM key-value heuristics (DICOM fields that are mapped to the BIDS labels)
# ------------------------------------------------------------------------------
  subject: <<filepath:/sub-(.*?)/>>   # This property extracts the subject label from the source directory
  session: <<filepath:/ses-(.*?)/>>   # This property extracts the session label from the source directory

  anat:       # ---------------------- All anatomical runs --------------------
  - provenance: /data/raw/sub-059/002-t1_mprage_sag_p2_iso_1.0/00001.IMA
    properties:
      filepath: ''
      filename: ''
      filesize: ''
      nrfiles: ''
    attributes:
      ProtocolName: t1_mprage_sag_p2_iso_1.0
      MRAcquisitionType: 3D
      Modality: MR
      SeriesDescription: t1_mprage_sag_p2_iso_1.0
      ImageType: "['ORIGINAL', 'PRIMARY', 'M', 'ND', 'NORM']"
      SequenceName: .tfl3d1_16ns
      SequenceVariant: "['SK', 'SP', 'MP']"
      ScanningSequence: "['GR', 'IR']"
      SliceThickness: '1'
      FlipAngle: '8'
      EchoNumbers: 1
      EchoTime: '3.03'
      RepetitionTime: '2300'
      PhaseEncodingDirection: ''
    bids:
      acq: t1mpragesagp2iso10
      ce:
      rec:
      run: <<1>>
      part: ['', mag, phase, real, imag, 0]
      suffix: T1w
    meta:
      DistortionCorrection: ND
      Comments: <<PatientComments>>
  - provenance: /data/raw/sub-059/003-mp2rage/00002.IMA
    properties:
      filepath: ''
```


FIGURE 4 | A snippet of study bidsmap in the YAML format. The bidsmap contains separate sections for each source data format (here "DICOM") and subsections for the BIDS data types (here "anat"). The provenance field contains the pathname of a source data sample that is representative of the run-item. The provenance data is not strictly necessary but very useful for a deeper inspection of the source data and for back-tracing, e.g., in case of encountering unexpected results. The arrow illustrates how the "properties" and "attributes" input dictionaries are mapped onto the "bids" and "meta" output dictionaries. This BIDS mapping together with the provenance item, i.e., the run-item, is the fundamental building block of a bidsmap. Note that the "part" value in the bids dictionary is a list, which is presented in the bidseditor GUI as a drop-down menu (with the first empty item being selected). Also, note that the special double bracket dynamic values (`<<1>>` and `<<PatientComments>>`) are explained in section "Dynamic Values."

applied to any dynamic property or attribute key. For instance, `{subject: <<PatientName:ID_(.*?)_>>}` would likewise have extracted "003" if the DICOM attribute PatientName was, e.g., "ID_003_anon." To test out dynamic values (either with or without appended regular expressions), users can handily enter them in the bidseditor within single brackets to instantly obtain their resulting value.

Dynamic values can handle many use cases and can be used throughout BIDScoin. Yet, two exceptions in the output dictionaries cannot always be handled directly with dynamic values. The first exception is the "run" index in the bids dictionary as this index cannot usually be determined from the data file alone. In that case, if the run-index is a dynamic number (e.g., `{run: <<1>>}`) and another output file with that run-index already exists, then during bidscoiner runtime this number will be incremented in compliance with BIDS standard (e.g., `{run: 2}`). If the run index is encoded in the header or file name, then the index can unambiguously be extracted using dynamic values. For instance, using `{run: <<ProtolName:run-(.*?)_>>}` will extract "3" if the DICOM ProtocolName is "`t1_mprage_sag_run-3_iso_1.0`." The second exception not covered by dynamic values is the "IntendedFor"[22] value in the meta dictionary, which also depends on the presence of other output files. Researchers can therefore specify IntendedFor images using a dynamic value with Unix shell-style wildcards. The bidscoiner will use these wildcards to lookup the appropriate images on disk. For instance, using `{IntendedFor: <<task>>}` will select all functional runs in the BIDS subject[/session] folder (as these runs always have "task" in their file name), and using

[22]IntendedFor values are used in BIDS to semantically indicate the association of an acquired data file with other acquired data files. A prominent example is a field-map that is acquired to correct fMRI images.

`{IntendedFor: <<Stop*Go><Reward>>}` will select all "Stop1Go"-, "Stop2Go"-, and "Reward"-runs.

The Plugin Interface for Interacting With Source Data

The BIDS community is working continuously to further improve and expand BIDS standard with new data types. Architecturally, to facilitate the implementation of such developments, all interactions of BIDScoin routines with the source data are done *via* a plugin layer that interacts in a data format-independent way. This paragraph describes the requirements and structure of plugins to allow advanced users and developers to write their own plugin and extend or customize BIDScoin to their needs. A BIDScoin plugin is a Python module with the following programming interface (functions):

- test(): A function to test the plugin and its options (see section "User Options").
- is_sourcefile(): A function to assess whether a source file is supported by the plugin. The return value should correspond to a data format section in the bidsmap.
- get_attribute(): A function to read an attribute value from a source file.
- bidsmapper_plugin(): A function to discover BIDS mappings in a source data session. To avoid code duplications and minimize plugin development time, various support functions are available to the plugin programmer in BIDScoin's library module named as "bids."
- bidscoiner_plugin(): A function to convert a single source data session to bids according to the specified BIDS mappings. Various support functions are available in the "bids" library module.

Each plugin has its own section in a bidsmap to store and edit its discovered BIDS mappings. Plugins can be installed by the user but the plugins described in the following sections come pre-installed.

Dcm2niix2bids: A Plugin for DICOM and PAR/XML Data

The "dcm2niix2bids" plugin is a wrapper around the well-known pydicom (Mason et al., 2020), nibabel (Brett et al., 2020), and dcm2niix tools (Li et al., 2016) for interacting with and converting the DICOM and Philips PAR(/REC)/XML source data. Pydicom is used to read DICOM attributes, nibabel is used to read PAR/XML attribute values, and dcm2niix is used to convert the DICOM and PAR/XML source data to NIfTI[23] and create BIDS sidecar files. These sidecar files contain standard metadata but, to give more control to the user, this metadata is appended or overwritten by the user data in the BIDS mapping meta dictionary. Dcm2niix2bids expects the source data files to be organized in:

- A "Series" subfolder organization. A Series folder is a subject[/session][24]-subfolder that contains files of a single data type, which are typically acquired in a single run – a.k.a "Series" in the DICOM standard. This format is often used by researchers in academic centers.
- A "DICOMDIR" organization with a DICOMDIR file in a single subject[/session] folder. A DICOMDIR is a dictionary file that indicates various places in a folder hierarchy of the available DICOM files. DICOMDIRs are often used in clinical settings.
- A flat DICOM organization. In a flat DICOM organization, all the DICOM files of all of the different Series are stored on a single subject[/session] folder. This organization is sometimes used when exporting data in clinical settings.
- A "PAR/XML" organization. All PAR/XML files of all the different Series in one folder. This organization is how users often export their data from Philips scanners in research settings (the session subfolder is optional): The PAR/XML session-data is expected to be organized in a single subject[/session] folder.

Spec2nii2bids: A Plugin for MR Spectroscopy Data

The "spec2nii2bids" plugin is a wrapper around the recent spec2nii[25] Python library for interacting with and converting the MR spectroscopy source data. Presently, the spec2nii2bids plugin is the first implementation that supports the conversion to BIDS for Philips SPAR/SDAT files, Siemens Twix files, and GE P-files. As with the dcm2niix2bids plugin, the produced sidecar files already contain the standard metadata that is complemented or overruled by the metadata that users specified in the bidseditor. Also, spec2nii2bids expects the source data to be organized in subject[/session] folders.

Phys2bidscoin: A Plugin for Physiological Data

The "phys2bidscoin" plugin is a wrapper around the phys2bids Python library (The phys2bids developers et al., 2019) for interacting with and converting physiological source data. Phys2bids currently supports the conversion of labchart (ADInstruments, Sydney, Australia) and AcqKnowledge (BIOPAC, Goleta, CA, United States) source files to compressed tab-separated value (".tsv.gz") files and create their json sidecar files, as per BIDS specifications. As in the other plugins, the sidecar files contain the standard metadata that is overwritten by the user data entered in the bidseditor. Phys2bidscoin expects the source data files to be organized in subject[/session] folders. This plugin has been developed during the OHBM hackathon 2021 and is still considered experimental at the moment of writing.

User Options

A bidsmap contains a separate "Options" section with dictionaries for the options and settings of BIDScoin and its plugins. Only plugins that are listed in the Options will be used by the BIDScoin routines, which allow researchers to use different plugins for different data sets. The template bidsmap Options are taken as default and can be adjusted using the bidseditor (**Figure 5**).

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

[24] Brackets indicate that this subfolder is optional.

[25] https://github.com/wtclarke/spec2nii.

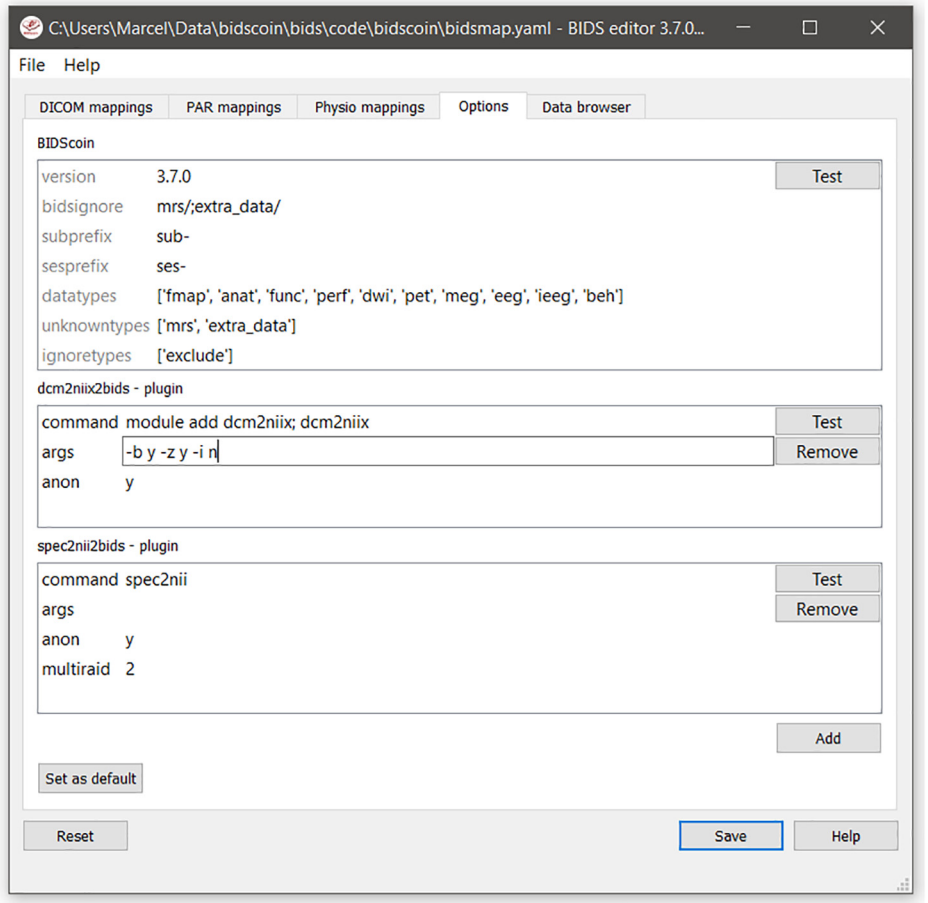


FIGURE 5 | The bidsmap options for BIDScoin and its plugins. Note that how the GUI automatically adapts with a new "Physio" tab due to the presence of physiological data in the repository, i.e., a "Physio" section in the study bidsmap. The user can manage the plugins that will be used with the "Add" and "Remove" buttons, and save the current options to the template bidsmap using the "Set default" button.

The BIDScoin Options

- **version:** Used to check for version conflicts between the installed version and the original version in the bidsmap (e.g., when upgrading the software after creating the bidsmap) or between the installed version and the latest online version.
- **bidsignore:** A semicolon-separated list of non-BIDS data types that are added to the .bidsignore file.
- subprefix: The subject prefix of the source data, e.g., "sub-."
- sesprefix: The session prefix of the source data, e.g., "ses-."
- datatypes: The list of data types that are converted to BIDS.
- unknowntypes: The list of data types that are converted to BIDS-like data type folders.
- ignoretypes: The list of data types that are excluded/not converted to BIDS.

The Dcm2niix2bids Plugin Options

- **command:** The command to run dcm2niix on the user system, e.g., "module add dcm2niix; dcm2niix."
- **args:** Argument string that is passed to dcm2niix to customize its behavior, e.g., −z n −i y for ignoring the derived data and having the uncompressed output data. The [Test] button can be used to test the proper working of the plugin.
- **anon:** Anonymization option (y/n) to round off age and discard acquisition date from the metadata.

The Spec2nii2bids Plugin Options

- command: The command to run spec2nii on the user system, e.g., "module add spec2nii; spec2nii."
- args: Argument string that is passed to spec2nii to customize its behavior.
- anon: Anonymization option (y/n) to round off age and discard acquisition date from the metadata.
- multiraid: A spec2nii (mapVBVD) argument for selecting the multiraid Twix file to load (default 2).

The Phys2bidscoin Plugin Options

The options and settings of this plugin are still under development.

RESULTS

BIDScoin has been developed in the Donders Institute for Cognitive Neuroimaging at the Radboud University. A large number of researchers inside and outside this institute have successfully used BIDScoin to convert their data sets to BIDS, exposing it to a wide range of source data formats, data types, data organizations, experimental paradigms, and equipment manufacturers. In this paper, we present a full workflow using tutorial MRI data. The goal of this tutorial is to demonstrate BIDScoin's functionality using the data that is representative of what researchers acquire in a standard neuroimaging experiment.

Tutorial MRI Data

The following steps are part of a tutorial that allows users to download phantom MRI data, to test the complete BIDScoin workflow, and to compare it to a reference output. It is assumed that BIDScoin 3.7 is installed and that the path string for "dcm2niix" in the template bidsmap "Options" section has been set correctly (see section "The Dcm2niix2bids Plugin Options").

Data Preparation

Before we can launch the GUI application and convert the data, we need to obtain a minimally organized source data set. In a shell terminal, create a tutorial playground folder by executing these commands:

```
$ bidscoin --download .    # Download the tutorial data
(use a "." for the current folder or adapt it to your
needs)
$ cd bidscointutorial    # Go to the downloaded data
(or provide the path to the subfolders when calling
the bidscoin tools)
```

The new "bidscointutorial" folder contains a "raw" source data folder and a "bids_ref" reference BIDS folder. The aim of this tutorial is to reproduce this bids_ref data folder. In the raw folder, these DICOM series (aka "runs") will be found:

- **001-localizer_32ch-head**: A localizer scan that is not scientifically relevant and can be left out of the BIDS data set.
- **002-AAHead_Scout_32ch-head**: A localizer scan that is not scientifically relevant and can be left out of the BIDS data set.
- **007-t1_mprage_sag_ipat2_1p0iso**: An anatomical T1-weighted scan.
- **047-cmrr_2p4iso_mb8_TR0700_SBRef**: A single-band reference scan of the subsequent multi-band functional MRI scan.
- **048-cmrr_2p4iso_mb8_TR0700**: A multi-band functional MRI scan.
- **049-field_map_2p4iso**: The field-map magnitude images of the first and second echo. Set as "magnitude1," bidscoiner will recognize the format. This field-map is intended for the previous functional MRI scan.
- **050-field_map_2p4iso**: The field-map phase difference image of the first and second echo.
- **059-cmrr_2p5iso_mb3me3_TR1500_SBRef**: A single-band reference scan of the subsequent multi-echo functional MRI scan.
- **060-cmrr_2p5iso_mb3me3_TR1500**: A multi-band multi-echo functional MRI scan.
- **061-field_map_2p5iso**: Idem, the field-map magnitude images of the first and second echo, intended for the previous functional MRI scan.
- **062-field_map_2p5iso**: Idem, the field-map phase difference image of the first and second echo.

Start with inspecting the raw data:

- Are the DICOM files for all the "bids/sub-*" folders organized in series-subfolders (e.g., "sub-001/ses-01/003-T1MPRAGE/0001.dcm," etc.)? BIDScoin's "dicomsort" utility can be used if this is not the case (hint: for didactical reasons this is not the case for sub-002). A help text for all BIDScoin tools is available by running the tool with the "-h" flag (e.g., "`rawmapper -h`").
- The "rawmapper" utility can be used to print out the DICOM values of the "EchoTime," "Sex," and "AcquisitionDate" of the fMRI series in the "raw" folder.

Brain Imaging Data Structure Mapping

Now, a study bidsmap can be made, i.e., the mapping from DICOM source files to BIDS target files. To that end, scan all folders in the raw data collection by running this "bidsmapper" command:

```
$ bidsmapper raw bids
```

- In the GUI that appears at the end, edit the task and acquisition labels of the functional scans into something more readable, e.g., "task-Reward" for the "acq-mb8" scans and "task-Stop" for the "acq-mb3me3 scans." Also, make the name of the T1 scan more user-friendly, e.g., by naming the acquisition label simply "acq-mprage."
- Add a search pattern to the "IntendedFor" field such that the first field-map will select the "Reward" runs and the second field-map the "Stop" runs.
- Since for this data set, we only have one session per subject, remove the session label (and note how the output names simplify, omitting the session subfolders and labels).
- When all done, go to the "Options" tab and change the "dcm2niix" settings to get the uncompressed NIfTI output data (i.e., "*.nii" instead of "*.nii.gz"). Test the tool to see if it can run and, as a final step, save the study bidsmap. Close the editor and re-edit the study bidsmap by running: `$ bidseditor bids`. See what happens if you remove the compulsory task label of a functional scan or if you enter values in the output dictionaries that are not BIDS-compliant, such as non-alphabetic characters.

Brain Imaging Data Structure Coining

The next step, converting the source data into a BIDS collection, is straightforward and can be repeated whenever the new data has come in. To convert the data, run the "bidscoiner" command-line tool (note that the input is the same as for the bidsmapper):

```
$ bidsmapper raw bids
```

- Check the "bids/code/bidscoin/bidscoiner.log" file and note that it contains the complete terminal output. Check the "bids/code/bidscoin/bidscoiner.errors" file and see if any warnings or errors did occur.
- Compare the results in the "bids/sub-*" subject folders with the in "bids_ref" reference result. Are the file and folder names the same (ignore the multi-echo and "extra_data" images)? Also check the json sidecar files of the field-maps. Do they have the right "EchoTime" and "IntendedFor" fields?
- Re-run the bidscoiner command. Are the same subjects processed again? Forcefully re-run "sub-001."

Finishing Up

Once the source data has been converted to BIDS, one still needs to do some additional work to make it ready for data analysis and sharing.

- Inspect the "participants.tsv" file and decide if it is ok.
- Update the "dataset_description.json" and "README" files.
- Combine the echoes using the "echocombine" tool, such that the individual echo images are replaced by the echo-combined image.
- Deface the anatomical scans using the "deface" tool. This will take a while but will obviously not work as normal for the (faceless) tutorial phantom data set. Therefore, store the "defaced" output in the "derivatives" folder (instead of, e.g., overwriting the existing images).
- As a final step, run the bids-validator[26] on your "bids_tutorial" folder. Is the BIDS repository now ready to be shared?

DISCUSSION

Brain Imaging Data Structure standard is paving the way for more sharing of neuroimaging data that can efficiently be processed in a standardized manner. In this paper, we have demonstrated the use case for and main working of our flexible and user-friendly BIDScoin application that can convert a variety of raw neuroimaging data formats to the latest BIDS version 1.6. BIDScoin adopts an intelligent mapping strategy to discover and convert source data as opposed to using programmatic logic. BIDScoin is designed to make as much use as possible of the information available on disk, i.e., the file properties and attributes, as well as of the information that can be retrieved directly from the user.

An important part in the workflow is a step in which the user can edit the resulting output file names and add additional metadata. This, in itself, is not a trivial task as BIDS standard is ever increasing and many of its entities are not self-explanatory. The bidseditor GUI is therefore equipped with many help functions, such as help texts, tooltips, visual cues, informative pop-up windows, field input validations, reset buttons, and data inspection windows. In addition, users can consult the online documentation or, for instance, ask questions on the GitHub BIDScoin issue page. The user-friendliness of applications such as BIDScoin is important as it reduces the amount of non-scientific work in the scientific process and, hence, allows neuroscientists to devote more time and energy to address their research questions of interest.

Advantages of BIDScoin

BIDScoin is a flexible framework for various reasons. It is written in Python and packaged and publicly released to pypi, hence the installation on multiple platforms is supported, including Linux, Windows, and macOS. Architecturally, BIDScoin makes the use of installable plugins, which increases the user-facing flexibility (e.g., to non-MRI data) and decreases the programmer-facing development costs. Researchers can modify or create their own plugins for specific data types without having to modify BIDScoin. As the entire framework is free and open-source, users are welcome and encouraged to contribute in this way.

Another feature contributing to BIDScoin's flexibility is an option to use regular expressions in the bidsmap, which are well known for their powerful string-searching algorithms and usage in many programming languages. Nevertheless, researchers normally do not need to know about or interact with regular expressions as these are typically used in the template bidsmap and are already created by advanced users or developers.

Different researchers and research institutes use different data acquisition and management conventions. To accommodate for this, BIDScoin users can customize the data discovery intelligence in the default template bidsmap and reduce the number of edits they need to make in the GUI. Such customization can be as simple as changing the attribute or property strings to reflect their prior knowledge about the data – a task that does not require any programming knowledge from the user.

BIDScoin errors, warnings, and normal operations on the bidsmap or on the data are printed in a standard human readable format in the terminal and simultaneously stored in logfiles in the BIDS output folder. The study bidsmap itself, with all its mapping values and options, is also stored in the BIDS output folder. The provenance of the data and its conversion to BIDS are therefore always searchable, verifiable, and reproducible.

Converting source data to BIDS cannot always be done using a single application. BIDScoin only adds data and does not delete or overwrite the existing data (unless the user specifies so) and is therefore safe to use in conjunction with other BIDS applications.

Limitations

While BIDScoin offers a convenient and flexible infrastructure for converting source data to BIDS, there are a few limitations to consider.

First, BIDScoin uses regular expressions instead of programmatic logic to map out source data. While this is a main feature, it also comes with the drawback that in certain situations the list of BIDS mappings in the bidsmap can become quite long and somewhat labor intensive to maintain

[26]https://bids-standard.github.io/bids-validator.

or edit. This may become apparent when researchers have very irregular ways to acquire the data, such as manually entered file names that vary slightly. Moreover, in a mapping approach, it may be more difficult to deal with exceptional sessions in which certain runs need to be treated differently from others. In those situations, users may need to write a (small) plugin to solve such cases programmatically.

Second, BIDScoin has initially been developed with MRI data in mind. This means that the current support for other source data formats is not as mature as that of MRI, or not (yet) present. Researchers may therefore need to do post-processing with additional software to obtain a fully converted BIDS-compliant data set. A common example that is not handled by BIDScoin is the conversion to BIDS of stimulus presentation logfiles. This conversion is difficult to automate in a generic way as the logfiles typically vary between experimental paradigms and researchers.

Third, while the BIDScoin GUI provides an easy way for researchers to add their knowledge about the different data types to the BIDS output folder, it does not do so for the few modality agnostic files, such as the "dataset_description.json" file in the root of the BIDS folder. BIDScoin creates these files with placeholder content if they are not present already, but users still need to open these files with a text editor afterward to add content.

Finally, BIDScoin requires a minimally organized source data repository with a subject[/session] folder structure. Although this is very common practice, some researchers may have a different organization or use data management solutions such as PACS (Choplin, 1992), XNAT (Marcus et al., 2007), or DataLad (Halchenko et al., 2021). In those cases, researchers may need to export or reorganize their source data or write a custom plugin before they can use BIDScoin.

Conclusion and Future Developments

BIDScoin is a new free and open-source framework for converting source data to BIDS. Its main features are flexibility and user-friendliness, that facilitate further adoption of BIDS standard, thus promoting data sharing and reproducibility. Currently, a plugin for physiological recordings is implemented and under testing, and a new PET plugin is under development. However, as the BIDS community and standard are continuously expanding, there is a need to develop more plugins to support more data formats and interface with data management solutions such as DataLad. With such a growing codebase in mind, it is important to grow a larger community of BIDScoin developers and to improve quality control by increasing the code coverage of the automated tests. An additional planned development is to release containerized versions of the software (Nichols et al., 2017) to deal with potentially increasingly complex dependencies and ensure exact reproducibility. At present, there is already a configuration file for Linux users to build their own BIDScoin Singularity container.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: SURFdrive, https://surfdrive.surf.nl/files/index.php/s/HTxdUbykBZm2cYM/download.

AUTHOR CONTRIBUTIONS

All authors contributed to the manuscript and approved the submitted version. MZ was the creator, main designer, and main code contributor of BIDScoin, and main writer of the manuscript. SM contributed to the phys2bidscoin plugin code and was a co-writer of the manuscript. RO contributed to the design of BIDScoin and was a co-writer of this manuscript.

ACKNOWLEDGMENTS

We would like to thank Rutger van Deelen for providing the initial (PyQt) setup and implementation of the bidseditor application and Yorguin José Mantilla Ramos for the useful architectural feedback and for the initial code of the sova2coin EEG/MEG plugin. We are also grateful for all the feedback, questions, and contributions that users have submitted on GitHub.

REFERENCES

Appelhoff, S., Sanderson, M., Brooks, T., Vliet, M., Quentin, R., Holdgraf, C., et al. (2019). MNE-BIDS: organizing electrophysiological data into the BIDS format and facilitating their analysis. *J. Open Source Softw.* 4:1896. doi: 10.21105/joss.01896

Brett, M., Markiewicz, C., Hanke, M., Côté, M., Cipollini, B., McCarthy, P., et al. (2020). *Nibabel*. Geneva: Zenodo, doi: 10.5281/zenodo.4295521

Choplin, R. (1992). Picture archiving and communication systems: an overview. *Radiographics* 12, 127–129. doi: 10.1148/radiographics.12.1.1734458

Gorgolewski, K., Auer, T., Calhoun, V., Cameron Craddock, R., Das, S., Duff, E., et al. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci. Data* 3:160044. doi: 10.1038/sdata.2016.44

Gorgolewski, K. J., Alfaro-Almagro, F., Auer, T., Bellec, P., Capotă, M., Chakravarty, M., et al. (2017). BIDS apps: improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. *PLoS Comput. Biol.* 13:e1005209. doi: 10.1371/journal.pcbi.1005209

Halchenko, Y., Goncalves, M., Visconti di Oleggio, Castello, M., Ghosh, S., Salo, T., et al. (2020). *A Flexible DICOM Converter for Organizing Brain Imaging Data Into Structured Directory Layouts*. Geneva: Zenodo, doi: 10.5281/zenodo.4390433

Halchenko, Y., Meyer, K., Poldrack, B., Solanky, D., Wagner, A., Gors, J., et al. (2021). DataLad: distributed system for joint management of code, data, and their relationship. *J. Open Source Softw.* 6:3262. doi: 10.21105/joss.03262

Holdgraf, C., Appelhoff, S., Bickel, S., Bouchard, K., D'Ambrosio, S., David, O., et al. (2019). iEEG-BIDS, extending the brain imaging data structure specification to human intracranial electrophysiology. *Sci. Data* 6:102. doi: 10.1038/s41597-019-0105-7

Knudsen, G. M., Ganz, M., Appelhoff, S., Boellaard, R., Bormans, G., Carson, R. E., et al. (2020). Guidelines for content and format of PET brain data in publications and in archives: a consensus paper. *J. Cereb. Blood Flow Metab.* 40, 1576–1585. doi: 10.1177/0271678X20905433

Li, X., Morgan, P. S., Ashburner, J., Smith, J., and Rorden, C. (2016). The first step for neuroimaging data analysis: DICOM to NIfTI conversion. *J. Neurosci. Methods* 264, 47–56. doi: 10.1016/j.jneumeth.2016.03.001

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. doi: 10.1385/ni:5:1:11

Mason, D., scaramallion, rhaxton, mrbean-bremen, Suever, J., Vanessasaurus, et al. (2020). *Pydicom: An Open Source DICOM Library*. Geneva: Zenodo, doi: 10.5281/zenodo.4313150

Moreau, C. A., Jean-Louis, M., Blair, R., Markiewicz, C. J., Turner, J. A., Calhoun, V. D., et al. (2020). The genetics-BIDS extension: easing the search for genetic data associated with human brain imaging. *GigaScience* 9:giaa104. doi: 10.1093/gigascience/giaa104

Nichols, T. E., Das, S., Eickhoff, S. B., Evans, A. C., Glatard, T., Hanke, M., et al. (2017). Best practices in data analysis and sharing in neuroimaging using MRI. *Nat. Neurosci.* 3, 299–303. doi: 10.1038/nn.4500

Niso, G., Gorgolewski, K. J., Bock, E., Brooks, T. L., Flandin, G., Gramfort, A., et al. (2018). MEG-BIDS, the brain imaging data structure extended to magnetoencephalography. *Sci. Data* 5:180110. doi: 10.1038/sdata.2018.110

Pernet, C. R., Appelhoff, S., Gorgolewski, K. J., Flandin, G., Phillips, C., Delorme, A., et al. (2019). EEG-BIDS, an extension to the brain imaging data structure for electroencephalography. *Sci. Data* 6:103. doi: 10.1038/s41597-019-0104-8

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

The phys2bids developers, D., Ayyagari, A., Bright, M., Ferrer, V., Gaudes, C. C., et al. (2019). *physiopy/phys2bids: BIDS Formatting of Physiological Recordings*. Geneva: Zenodo, doi: 10.5281/zenodo.3586045

Turner, J., Eickhoff, S., and Nichols, T. (2016). Sharing the wealth: brain imaging repositories in 2015. *NeuroImage* 124, 1065–1262.

White, T., Blok, E., and Calhoun, V. D. (2020). Data sharing and privacy issues in neuroimaging research: opportunities, obstacles, challenges, and monsters under the bed. *Hum. Brain Mapp.* 1–14. doi: 10.1002/hbm.25120 [Epub ahead of print].

Conflict of Interest: 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Neuroinformatics

TECHNOLOGY AND CODE
published: 28 January 2022
doi: 10.3389/fninf.2021.767936

PyNeval: A Python Toolbox for Evaluating Neuron Reconstruction Performance

Han Zhang[1,2], *Chao Liu*[1,2], *Yifei Yu*[3], *Jianhua Dai*[4], *Ting Zhao*[5]* and *Nenggan Zheng*[1,3,4]*

[1] Qiushiq Academy for Advanced Studies (QAAS), Zhejiang University, Hangzhou, China, [2] College of Computer Science and Technology, Zhejiang University, Hangzhou, China, [3] Zhejiang Lab, Hangzhou, China, [4] Collaborative Innovation Center for Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU), Hangzhou, China, [5] Howard Hughes Medical Institute, Janelia Research Campus, Ashburn, VA, United States

Quality assessment of tree-like structures obtained from a neuron reconstruction algorithm is necessary for evaluating the performance of the algorithm. The lack of user-friendly software for calculating common metrics motivated us to develop a Python toolbox called PyNeval, which is the first open-source toolbox designed to evaluate reconstruction results conveniently as far as we know. The toolbox supports popular metrics in two major categories, geometrical metrics and topological metrics, with an easy way to configure custom parameters for each metric. We tested the toolbox on both synthetic data and real data to show its reliability and robustness. As a demonstration of the toolbox in real applications, we used the toolbox to improve the performance of a tracing algorithm successfully by integrating it into an optimization procedure.

Keywords: PyNeval, metric, quantitative analysis, neuron tracing, neuron reconstruction, toolbox

OPEN ACCESS

Edited by:
Andrew P. Davison,
UMR9197 Institut des Neurosciences Paris Saclay (Neuro-PSI), France

Reviewed by:
Hua Han,
Institute of Automation, Chinese Academy of Sciences (CAS), China
John McAllister,
Queen's University Belfast, United Kingdom

***Correspondence:**
Ting Zhao
zhaot@janelia.hhmi.org
Nenggan Zheng
zng@zju.edu.cn

Received: *31 August 2021*
Accepted: *27 December 2021*
Published: *28 January 2022*

Citation:
Zhang H, Liu C, Yu Y, Dai J, Zhao T and Zheng N (2022) PyNeval: A Python Toolbox for Evaluating Neuron Reconstruction Performance. Front. Neuroinform. 15:767936. doi: 10.3389/fninf.2021.767936

1. INTRODUCTION

Reconstructing tree structures of labeled neurons in light microscope images is a critical step for neuroscientists to study neural circuits (Parekh and Ascoli, 2013; Peng et al., 2015). Researchers have longed for automating this process of neuron reconstruction, also called neuron tracing, to overcome the bottleneck of manual annotation or proofreading (Gillette et al., 2011b; Peng et al., 2011). Despite decades of efforts (Halavi et al., 2012; Acciai et al., 2016), however, there is still no computer algorithm that can be as reliable as human labor. Besides being a complex computer vision problem itself, neuron tracing has baffled developers on how an algorithm should be evaluated. Unlike many image segmentation problems, neuron tracing has no universally accepted metric to measure its performance. In fact, it is infeasible to design one metric for all applications, which have different tolerance to different types of reconstruction errors. The real problem here is a lack of easy access to evaluation metrics. As a result, researchers have to implement a metric by themselves or compromise on metric properness for convenience. This has caused two issues in the literature. First, performance evaluation was often limited to one or two metrics that were not sufficient to offer comprehensive comparisons. Second, the metrics applied were ambiguous in general without open implementations, causing potential inconsistency and low reproducibility.

This problem can be addressed by open-source user-friendly software that allows evaluating neuron reconstruction qualities in various ways. Such software should cover the two major categories of reconstruction metrics, geometrical metrics and topological metrics. Geometrical metrics measure how well a reconstructed model overlaps with the underlying gold standard or ground truth model, while topological metrics measure the topological similarity between the two

models. Geometrical metrics are often computed by summarizing spatial matching between the two models, such as counting the number of matched nodes as done in the popular substantial spatial distance (SSD) metric (Peng et al., 2010) or measuring the length of overlapped branches in the so called length metric (Wang et al., 2011). These metrics are straightforward for telling where branches are missing or over-traced in reconstruction, but they are not suitable for evaluating topological accuracy, which is crucial in some applications such as electrophysiological simulation. For the latter situation, topological metrics such as the Digital Reconstruction of Axonal and Dendritic Morphology (DIADEM) metric (Gillette et al., 2011a), tree edit distance (Bille, 2005), and critical node (CN) metric (Feng et al., 2015) are preferred.

Hence, we introduce a Python toolbox called PyNeval, which is the first open-source toolbox designed to provide multiple choices for evaluating the qualities of reconstruction results conveniently. In specific, PyNeval is designed to have the following features:

- PyNeval has a user-friendly command-line interface for easy use and a flexible way of configuring parameters for covering a broad range of user requirements.
- PyNeval provides various evaluation methods for measuring both geometrical and topological qualities of reconstructions.
- PyNeval provides an interface for optimizing any reconstruction algorithm that converts an image into an SWC file with adjustable parameters.

In this paper, we formulate each evaluation method implemented in PyNeval under a mathematical framework if it has not been clearly defined in the literature. Our implementation follows those formulations, which give users a clear and unambiguous picture of what PyNeval computes. We apply PyNeval to randomly perturbed data to show that PyNeval can produce reliable evaluation scores from different metrics. The difference among the metrics can be seen in their results of manually-designed special cases. Besides comparing different tracing algorithms, PyNeval can be used to optimize any reconstruction algorithm with tunable parameters, as demonstrated in our experiment on mouse brain data acquired by fMOST (Gong et al., 2016).

2. METHOD

2.1. SWC Format

The PyNeval toolbox is designed based on the SWC format (Cannon et al., 1998), the common format of neuron reconstruction results. The format represents the shape of a neuron in a tree structure that consists of a set of hierarchically organized nodes (Feng et al., 2015):

$$T = \{\mathbf{n}_i = (\mathbf{x}_i, r_i, \mathbf{n}_j) \mid i = 1, ..., N_T, \mathbf{n}_j \in T \cup \mathbf{n}_0, i \neq j, \\ \mathbf{x}_i \in \mathbb{R}^3, r_i \in \mathbb{R}\} \quad (1)$$

where $N_T = |T|$ is the number of nodes of T, the ith node $\mathbf{n}_i$ is a sphere centering at $\mathbf{x}_i = (x_i, y_i, z_i)$ with radius r_i, and $\mathbf{n}_0$ is a virtual node. In this definition, $\mathbf{n}_j$ is called the parent of $\mathbf{n}_i$, and a node with a virtual node as its parent is called a root node. For convenience, we also define the following functions:

- Parent of a node: $\rho : (\mathbf{x}_i, r_i, \mathbf{n}_j) \in T \mapsto \mathbf{n}_j \in T \cup \mathbf{n}_0$
- Position of a node: $\mathbf{x} : (\mathbf{x}_i, r_i, \mathbf{n}_j) \in T \mapsto \mathbf{x}_i \in \mathbb{R}^3$
- Radius of a node: $r : (\mathbf{x}_i, r_i, \mathbf{n}_j) \in T \mapsto r_i \in \mathbb{R}$

The edge set of the model T is defined as

$$E(T) = \{\mathbf{e}_i \mid \mathbf{e}_i = (\mathbf{n}_i, \rho(\mathbf{n}_i)), \mathbf{n}_i \in T\} \quad (2)$$

One important constraint on the SWC model T is that the graph $G = (T \cup \mathbf{n_0}, E(T))$ has no loop, which means that it is a tree.

2.2. Software Design

Assuming that the reconstruction results are in the SWC format, PyNeval takes a gold standard SWC file as well as one or more testing SWC files and outputs the quality scores for each testing SWC. Since PyNeval supports multiple metrics, it should also allow the user to specify metric options. As a consequence, input SWC files and metric options form the essential parameters of the main PyNeval command. While this provides a straightforward interface for an application, it is not flexible enough to adapt to more subtle user requirements such as setting specific parameters for a certain metric or checking evaluation details. Therefore, PyNeval has a flexible but friendly way of accepting optional parameters, allowing the user to specify these parameters without having to check extensive documents. PyNeval can output carefully formatted results to the screen for easy reading or save the results with more details to a file for further analysis, depending on the user's choice of the output parameters. For example, the `-detail` option can be used to produce an SWC file that labels each node in the test structure with a specific type to indicate what kind of error is associated with that node. The overall architecture of PyNeval is shown in **Figure 1**.

2.3. Metrics

PyNeval supports four commonly used metrics in both geometrical and topological categories, although it can be easily extended to more metrics. To explain the metrics implemented in PyNeval unambiguously, we use the notations listed in **Table 1**.

More specifically, some of the notations can be interpreted as follows:

- Besides always assuming that $\mathbf{e}_{ij} = (\mathbf{n}_i, \mathbf{n}_j)$ and $\mathbf{e}_i = (\mathbf{n}_i, \rho(\mathbf{n}_i))$, we also use $\mathbf{n}$ and $\mathbf{e}$ to represent a node and an edge, respectively, when there is no need to index them.
- Node interpolation

$$\mathcal{I}(\mathbf{n}, \lambda) = \begin{cases} ((1-\lambda)\mathbf{x}(\mathbf{n}) + \lambda\mathbf{x}(\rho(\mathbf{n})), (1-\lambda)r(\mathbf{n}) \\ \quad + \lambda r(\rho(\mathbf{n})), \rho(\mathbf{n}))), & 0 \leq \lambda < 1 \\ \rho(\mathbf{n}), & \lambda = 1 \end{cases} \quad (3)$$

- Interpolation between two nodes, no matter if they are connected

$$\mathcal{I}(\mathbf{n}_i, \mathbf{n}_j, \lambda) = \begin{cases} ((1-\lambda)\mathbf{x}_i + \lambda\mathbf{x}_j, (1-\lambda)r_i \\ \quad + \lambda r_j, \mathbf{n}_j), & 0 \leq \lambda < 1 \\ \mathbf{n}_j, & \lambda = 1 \end{cases} \quad (4)$$

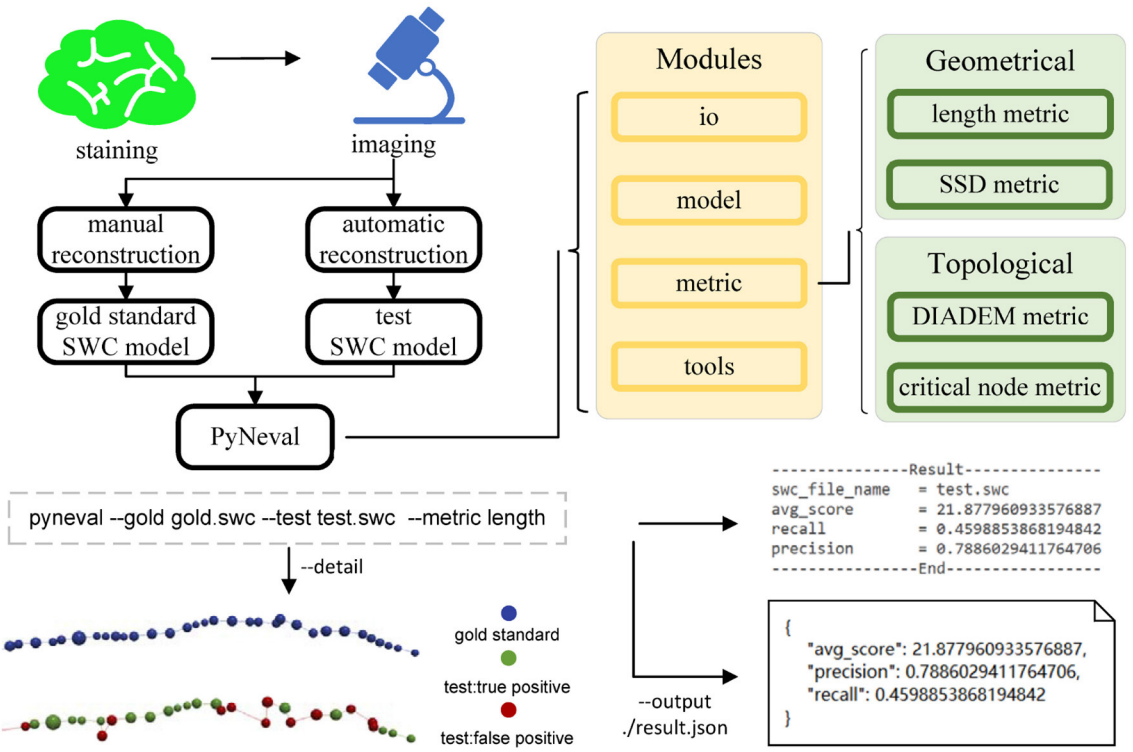


FIGURE 1 | The overview of PyNeval, which can take a gold standard model and a test model from the same image as input through a command line interface and outputs quality scores of the test model, as well as more details about reconstruction errors. Four different metrics, including the length and substantial spatial distance (SSD) metrics in the geometrical category and the DIADEM and critical node (CN) metrics in the topological category, are available in PyNeval.

TABLE 1 | Mathematical notations used for explaining the metrics.

Symbol	Meaning
T_g	Gold standard SWC model
T_t	SWC model for evaluation
$\mathbf{n}_i$	A node in a SWC model with an unique index i
$E(T)$	Set of all edges in T
$\mathbf{e}_i$	Edge from node $\mathbf{n}_i$ to node $\rho(\mathbf{n}_i)$
$d(x, y)$	Distance between two objects, which can be nodes, edges or trees
L	Length of an edge or an edge set
M	Matched node or edge set
$\mathcal{I}$	Interpolation function

- Node distances

$$d(\mathbf{n}_i, \mathbf{n}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{5}$$

$$d_{xy}(\mathbf{n}_i, \mathbf{n}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6}$$

$$d_z(\mathbf{n}_i, \mathbf{n}_j) = |z_i - z_j| \tag{7}$$

$$d(\mathbf{n}, \mathbf{e}_i) = \min_{\lambda} d(\mathbf{n}, \mathcal{I}(\mathbf{n}_i, \lambda)) \tag{8}$$

$$d(\mathbf{n}, T) = \min_{\mathbf{e} \in E(T)} d(\mathbf{n}, \mathbf{e}) \tag{9}$$

- Node length

$$L(\mathbf{n}) = \begin{cases} 0, & \mathbf{n} \text{ is a root node} \\ d(\mathbf{n}, \rho(\mathbf{n})), & \text{otherwise} \end{cases} \tag{10}$$

- Edge lengths

$$L(\mathbf{e}_{ij}) = d(\mathbf{n}_i, \mathbf{n}_j) \tag{11}$$

$$L(E(T)) = \sum_{\mathbf{e} \in E(T)} L(\mathbf{e}) \tag{12}$$

- Tree length

$$L(T) = L(E(T)) \tag{13}$$

2.3.1. Length Metric

It is natural to evaluate the quality of a reconstruction T_t by measuring how well its branches overlap with the gold standard model T_g. This can be computed by matching edges between T_t

and T_g and then summing up the lengths of the matched edges in T_t and T_g, respectively, to produce the common precision and recall metrics. Before proceeding to explain the length metric in detail, we first need to define some more notations

- A segment lying on an edge $(\mathbf{n}, \rho(\mathbf{n}))$ is

$$C(\mathbf{n}, \lambda_1, \lambda_2) = \{\mathbf{x}(\mathcal{I}(\mathbf{n}, \lambda)) | 0 \leq \lambda_1 \leq \lambda \leq \lambda_2 \leq 1\} \tag{14}$$

  and its length is $L(C(\mathbf{n}, \lambda_1, \lambda_2)) = (\lambda_2 - \lambda_1)L(\mathbf{n})$.
- The overlap ratio between two segments C_1, C_2 with respect to the edge $(\mathbf{n}, \rho(\mathbf{n}))$ is defined as $\mathcal{O}(C_1, C_2)$. Suppose that $C_1 = (\mathbf{n}_i, \alpha_1, \alpha_2)$, $C_2 = C(\mathbf{n}_j, \beta_1, \beta_2)$, the overlap ratio $\mathcal{O}(C_1, C_2)$ is

$$\mathcal{O}(C_1, C_2) = \begin{cases} \max(0, \min(\alpha_2 - \alpha_1, \\ \beta_2 - \beta_1, \alpha_2 - \beta_1, \beta_2 - \alpha_1)), & i = j \\ 0, & i \neq j \end{cases} \tag{15}$$

- A simple path between two points on a tree is

$$\mathcal{P}((\mathbf{n}_s, \lambda_s), (\mathbf{n}_t, \lambda_t)) = \begin{cases} \{C(\mathbf{n}_s, \min(\lambda_s, \lambda_t), \max(\lambda_s, \lambda_t))\}, & s = t \\ \{C(\mathbf{n}_{i_k}, \alpha_k, \beta_k) | k = 1 \cdots K\}, & s \neq t \end{cases} \tag{16}$$

  where $i_1 = s$, $i_K = t$, $\mathbf{n}_{i_{k-1}} = \rho(\mathbf{n}_{i_k})$ or $\mathbf{n}_{i_k} = \rho(\mathbf{n}_{i_{k-1}})$, K is the number of edges on the path and

$$(\alpha_k, \beta_k) = \begin{cases} (0, \lambda_s), & k = 1 \text{ and } \mathbf{n}_s = \rho(\mathbf{n}_{i_2}) \\ (\lambda_s, 1), & k = 1 \text{ and } \rho(\mathbf{n}_s) = \mathbf{n}_{i_2} \\ (0, \lambda_t), & k = K \text{ and } \mathbf{n}_t = \rho(\mathbf{n}_{i_{K-1}}) \\ (\lambda_t, 1), & k = K \text{ and } \rho(\mathbf{n}_t) = \mathbf{n}_{i_{K-1}} \\ (0, 1), & \text{otherwise} \end{cases} \tag{17}$$

In our implementation, we construct the matched edge set between T_t and T_g as demonstrated in **Algorithm 1**.

2.3.2. SSD Metric

The SSD metric (Peng et al., 2010) can be viewed as a variant of the length metric in terms of what it tries to measure. Instead of matching edges directly, however, SSD counts how many nodes are matched without excluding duplicated matches. Besides, SSD provides an additional metric to measure how far the unmatched nodes are away from the counterpart model. One extra step of SSD metric is resampling each branch of T_t and T_g uniformly to reach a sufficient density ε_{sp}

$$\mathcal{R}(T) = \{\mathbf{n}_k^{(i)} | \mathbf{n}_i \in T, k = 0, 1, \ldots, K_i\} \tag{18}$$

where $\mathbf{n}_k^{(i)} = \mathcal{I}(\mathbf{n}_i, \mathbf{n}_{k+1}^{(i)}, \frac{k}{k+1})$, $\mathbf{n}_{K_i}^{(i)} = \mathbf{n}_0^{(j)}$, $\rho(\mathbf{n}_i) = \mathbf{n}_j$, $K_i \varepsilon_{sp} \leq L(\mathbf{e}_{ij})$, and $(K_i + 1)\varepsilon_{sp} > L(\mathbf{e}_{ij})$.

After that, like computing the length metric, the SSD metric can be obtained by constructing the matched node set $M_{\mathbf{n}}$ between two SWC models T_g and T_t shown in **Algorithm 2**.

2.3.3. CN Metric

The CN metric measures how many CNs are reconstructed correctly. A critical node is either a branching or terminal node, which determines the topology of an SWC model. Mathematically, the set of the CNs of an SWC model T is defined as

$$\mathcal{K}(T) = \{\mathbf{n} | \mathbf{n} \in T, D_T(\mathbf{n}) \neq 2\} \tag{19}$$

where $D_T(\mathbf{n})$ is the degree of node $\mathbf{n}$ in the tree T.

Algorithm 1: Length metric.

Input: $T_g, T_t, \epsilon_l \in \mathbb{R}^+, \epsilon_o \in \mathbb{R}^+, \epsilon_d \in \mathbb{R}^+$
Output: precision, recall

$M_t \leftarrow \emptyset, M_g \leftarrow \emptyset$
for $\mathbf{e}_{ij}$ in $E(T_t)$ **do**
  $\mathcal{I}(\mathbf{n}'_{g_1}, \lambda_1) \leftarrow \arg\min_{\mathbf{n}' \in \cup_{\mathbf{n} \in T_g} \{\mathcal{I}(\mathbf{n}, \lambda) | 0 \leq \lambda \leq 1\}} d(\mathbf{n}_i, \mathbf{n}')$
  $\mathcal{I}(\mathbf{n}'_{g_2}, \lambda_2) \leftarrow \arg\min_{\mathbf{n}' \in \cup_{\mathbf{n} \in T_g} \{\mathcal{I}(\mathbf{n}, \lambda) | 0 \leq \lambda \leq 1\}} d(\mathbf{n}_j, \mathbf{n}')$
  if $\max(d(\mathbf{n}_i, \mathcal{I}(\mathbf{n}'_{g_1}, \lambda_1)), d(\mathbf{n}_j, \mathcal{I}(\mathbf{n}'_{g_2}, \lambda_2))) < \epsilon_d$ **then**
    $\mathcal{P}((\mathbf{n}'_{g_1}, \lambda_1), (\mathbf{n}'_{g_2}, \lambda_2))$ is the simple path between $\mathcal{I}(\mathbf{n}'_{g_1}, \lambda_1)$ and $\mathcal{I}(\mathbf{n}'_{g_2}, \lambda_2)$
    if $\frac{|L(\mathbf{e}_{ij}) - L(\mathcal{P}((\mathbf{n}'_{g_1}, \lambda_1), (\mathbf{n}'_{g_2}, \lambda_2)))|}{L(\mathbf{e}_{ij})} < \varepsilon_l$ and $\max_{C_1 \in \mathcal{P}((\mathbf{n}'_{g_1}, \lambda_1), (\mathbf{n}'_{g_2}, \lambda_2)), C_2 \in M_g} \mathcal{O}(C_1, C_2) < \varepsilon_o$ **then**
      $M_t \leftarrow M_t \cup \{\mathbf{e}_{ij}\}$
      $M_g \leftarrow M_g \cup \mathcal{P}((\mathbf{n}'_{g_1}, \lambda_1), (\mathbf{n}'_{g_2}, \lambda_2))$
    end if
  end if
end for
precision $\leftarrow \frac{L(M_t)}{L(T_t)}$
recall $\leftarrow \frac{L(M_g)}{L(T_g)}$
return precision, recall

Algorithm 2: SSD metric.

Input: $\mathcal{R}(T_g), \mathcal{R}(T_t), \epsilon_{sp} \in \mathbb{R}^+, \epsilon_{ssd} \in \mathbb{R}^+$
Output: precision, recall, SSD cost

$M_n(T_g, T_t) \leftarrow \emptyset, M_n(T_t, T_g) \leftarrow \emptyset$
for $\mathbf{n}_i$ in $\mathcal{R}(T_g)$ **do**
  if $\min_{\mathbf{n}_j \in \mathcal{R}(T_t)} d(\mathbf{n}_i, \mathbf{n}_j) < \varepsilon_{ssd}$ **then**
    $M_n(T_g, T_t) \leftarrow M_n(T_g, T_t) \cup \{\mathbf{n}_i\}$
  end if
end for

for $\mathbf{n}_i$ in $\mathcal{R}(T_t)$ **do**
  if $\min_{\mathbf{n}_j \in \mathcal{R}(T_g)} d(\mathbf{n}_i, \mathbf{n}_j) < \varepsilon_{ssd}$ **then**
    $M_n(T_t, T_g) \leftarrow M_n(T_t, T_g) \cup \{\mathbf{n}_i\}$
  end if
end for

precision $\leftarrow \frac{|M_n(\mathcal{R}(T_t), \mathcal{R}(T_g))|}{|\mathcal{R}(T_t)|}$
recall $\leftarrow \frac{|M_n(\mathcal{R}(T_g), \mathcal{R}(T_t))|}{|\mathcal{R}(T_g)|}$
SSD cost $\leftarrow \frac{SSD(\mathcal{R}(T_t), \mathcal{R}(T_g)) + SSD(\mathcal{R}(T_g), \mathcal{R}(T_t))}{2}$
return precision, recall, SSD cost

Algorithm 3: Critical node metric.

Input: $\mathcal{K}(T_g), \mathcal{K}(T_t), \epsilon_{br} \in \mathbb{R}^+$
Output: precision, recall
$V_b \leftarrow \mathcal{K}(T_t) \cup \mathcal{K}(T_g)$
$E_b \leftarrow \{(\mathbf{n}^{(t)}, \mathbf{n}^{(g)}) | \mathbf{n}^{(t)} \in \mathcal{K}(T_t), \mathbf{n}^{(g)} \in \mathcal{K}(T_g), d(\mathbf{n}^{(t)}, \mathbf{n}^{(g)}) < \varepsilon_{br}\}$
$G_b \leftarrow (V_b, E_b)$
$M_b^* \leftarrow \arg\max_{M_b} |M_b|$ #M_b is a matching in G_b, i.e., M_b is a subgraph of G_b and all of its nodes have degree 1.
precision $\leftarrow \frac{|M_b^*|}{|\mathcal{K}(T_t)|}$
recall $\leftarrow \frac{|M_b^*|}{|\mathcal{K}(T_g)|}$
return precision, recall

Algorithm 4: Diadem metric.

Input: $\mathcal{K}(T_g), \mathcal{K}(T_t), \epsilon_{xy} \in \mathbb{R}^+, \epsilon_z \in \mathbb{R}^+, , \epsilon_{ld} \in \mathbb{R}^+$
Output: DIADEM score
for $\mathbf{n}_i \in \mathcal{K}(T_g)$ **do**
  for $\mathbf{n}_j \in \mathcal{K}(T_t)$ **do**
    if $d_{xy}(\mathbf{n}_i, \mathbf{n}_j) < \epsilon_{xy}$ and $d_z(\mathbf{n}_i, \mathbf{n}_j) < \epsilon_z$ **then**
      # search for $\alpha(\mathbf{n})$, the ancestor of $\mathbf{n}$ on the path between $\mathbf{n}$ and its root $\mathbf{n}_0$.
      # $\mathbf{n}_0^{(g)}$, $\mathbf{n}_0^{(t)}$ are the roots of gold and test trees respectively.
      for $\alpha(\mathbf{n}_i)$ in $\mathcal{P}((\mathbf{n}_i, 0), (\mathbf{n}_0^{(g)}, 0))$ **do**
        for $\alpha(\mathbf{n}_j)$ in $\mathcal{P}((\mathbf{n}_j, 0), (\mathbf{n}_0^{(t)}, 0))$ **do**
          if $\alpha(\mathbf{n}_i)$ matches $\alpha(\mathbf{n}_j)$ and $\frac{|L(\mathcal{P}(\mathbf{n}_i, \alpha(\mathbf{n}_i))) - L(\mathcal{P}(\mathbf{n}_j, \alpha(\mathbf{n}_j)))|}{L(\mathcal{P}(\mathbf{n}_i, \alpha(\mathbf{n}_i)))} < \epsilon_{ld}$ **then**
            $M_d \leftarrow M_d \cup \{\mathbf{n}_i\}$
          end if
        end for
      end for
    end if
  end for
end for
DIADEM score $= \frac{\sum_{\mathbf{n} \in M_d} D_{T_g}(\mathbf{n})}{\sum_{\mathbf{n} \in \mathcal{K}(T_g)} D_{T_g}(\mathbf{n})}$
return DIADEM score

With the CNs, we can compute the CN metric with **Algorithm 3**.

2.3.4. DIADEM Metric

Introduced by Gillette et al. (2011a) for the DIADEM challenge (Gillette et al., 2011b), the DIADEM metric evaluates the similarity between two models by comparing their branching structures. Like the CN metric, the DIADEM metric is also based on matching CNs in $\mathcal{K}(T_g)$ and $\mathcal{K}(T_t)$, here $\mathcal{K}(T)$ is defined in equation (19). But its matching criteria are more complicated than simply checking the distances. A brief description of the DIADEM metric is proposed as **Algorithm 4**.

TABLE 2 | Summary of neuron reconstructions from six image stacks.

ID	Number of nodes	Number of roots	Source
BN1	4,966	7	BigNeuron
BN2	852	7	BigNeuron
BN3	432	2	BigNeuron
BN4	4,251	4	BigNeuron
FM1	5,160	63	fMOST
FM2	674	9	fMOST

There are also several rounds of scanning to deal with the problem that $\mathbf{n}_j \in \mathcal{K}(T_t)$ is not the only node that meets the conditions, and labels every unmatched CN in T_g as a match if it is on a matched path. More details can be found in the reference (Gillette et al., 2011a).

2.4. Implementation

PyNeval is implemented in Python 3 (Oliphant, 2007) using several powerful open-source packages, including Numpy (Van Der Walt et al., 2011) for numerical computation, Anytree (Anytree., 2020) for handling the SWC data structure, and kdtree as well as Rtree (KDtree., 2017; Rtree., 2020) for fast search of closest edges and nodes.

3. RESULTS

3.1. Robustness Test

We applied PyNeval to randomly perturbed gold standard reconstructions to characterize each metric and evaluate the robustness of our program. The perturbed dataset is constructed by randomly moving a portion of nodes in the original reconstructions, which are gold standard SWC models from the standard BigNeuron dataset (Peng et al., 2015) as well as our custom dataset acquired from fMOST (Gong et al., 2016). As listed in **Table 2**, a total of six reconstructions with a large variety of sizes were used for the test.

A reasonable metric should produce decreasing quality scores as the perturbation ratio increases. This can be seen in the experimental results plotted in **Figure 2**, in which each curve shows the trend of a metric score along the increasing perturbation ratio. Each metric score at a perturbation ratio was averaged from 10 trials for a sequence of 11 perturbation ratios increasing by the step of 0.1 from 0 to 1. As expected, the curves are consistently similar among different models, in spite of their different morphologies. They all follow the right trend that more perturbation results in a worse score. We can also see that, topological metrics have higher variance than geometrical metrics, which is not surprising because how a perturbation affects the topology highly depends on the positions of the perturbed nodes. This suggests that when we use a topological metric to evaluate an algorithm, more samples or trials might be needed to draw a reliable conclusion.

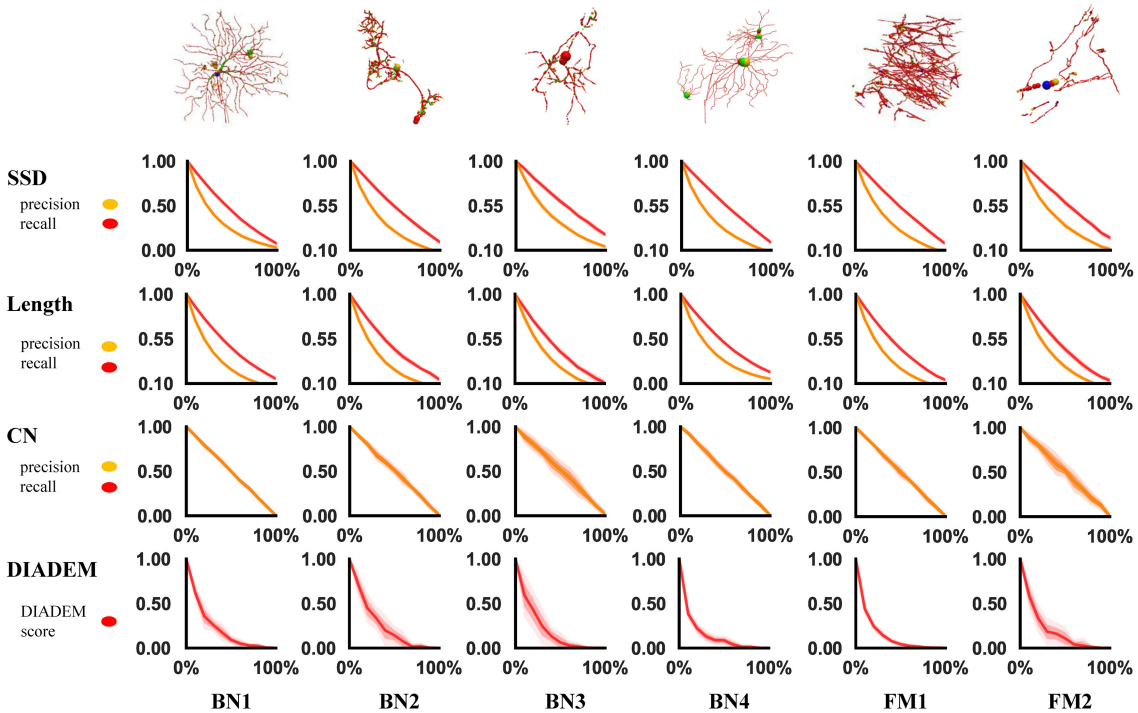


FIGURE 2 | Result of robustness test. Each row represents a metric and each column represents an swc model. In each chart, the x-axis is the perturbed proportion and the y-axis is the corresponding metric value.

3.2. Special Case Analysis

In addition to the perturbation experiment, we also tested the behaviors of the metrics on some special cases to show their differences more clearly. We constructed four special cases for geometrical metrics and the other four for topological metrics, including **Figure 3**:

- Test cases for geometrical metrics
  1. Both ends of an edge in T_g have matched nodes in T_t, but T_t has an extra node that deviates the path from the edge segment in T_g.
  2. T_g manages to find a match path in T_t, but its nodes do not match those on the same path in T_g due to the sampling rate.
  3. A straight path in T_g is reconstructed as a bifurcation in T_t by mistake.
  4. T_t distorts a relatively straight path in T_g into a zigzag path.
- Test cases for topological metrics
  1. The nodes are matched but a wrong connection in T_t changes the root to a non-CN.
  2. T_t has wrong connections, but all the CNs are still matched between T_t and T_g.
  3. The reconstruction moves a node and all its descendants to a different location.
  4. Connection mistakes in T_t break the original model into several isolated graphs.

Table 3 shows different results on the same special cases produced by the SSD and length metrics. The SSD metric tends to output higher F1 scores than the length metric does, but it is not necessarily better or worse. In some cases (**Figures 3A,B**), the SSD scores look more reasonable because their more granulated matching can capture partial matching of a path. In other cases (**Figures 3C,D**), where the errors are more complicated, however, the SSD metric can overestimate reconstruction qualities by counting duplicated matches.

The difference between the two topological metrics can be seen in **Figure 4** and **Table 4**. The CN metric fails to detect reconstruction errors in **Figures 4A,D** because the errors do not add or remove a critical node. The DIADEM metric can avoid such a problem by including path comparison. In this sense, the DIADEM metric is more comprehensive than the other three metrics in PyNeval as it actually considers both topological and geometrical features. Nevertheless, we should note that it may not correlate well with the amount of editing work needed to fix errors. For example, the test model in case 2 can be more readily fixed than case 3, despite that it has a lower DIADEM score. In other words, the DIADEM metric can be misleading when we expect an automatic method to minimize manual work.

3.3. Reconstruction Parameter Optimization Using PyNeval

Besides comparing different reconstruction algorithms, another important application of PyNeval is to optimize parameters of

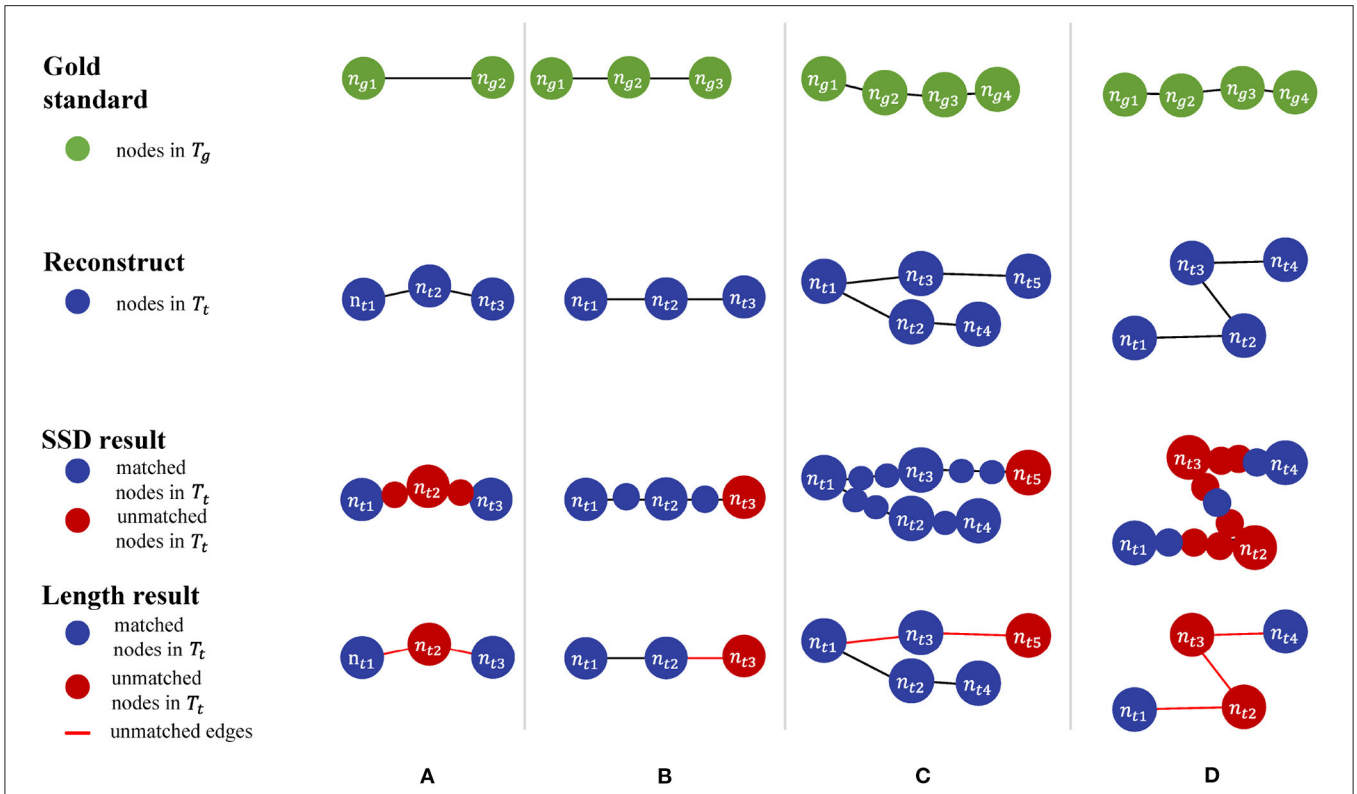


FIGURE 3 | (A–D) Four manually constructed cases for testing geometrical metrics.

TABLE 3 | PyNeval results of the SSD and length metrics for the geometrical cases are shown in **Figures 3A–D**.

Method	Index	File name			
		A	B	C	D
SSD metric	SSD score	1.66	1.60	0.51	1.49
	Recall	0.33	0.86	1.00	0.27
	Precision	0.36	0.83	0.95	0.18
	F1 score	0.35	0.85	0.98	0.21
Length metric	Recall	0.00	0.47	1.00	0.00
	Precision	0.00	0.50	0.54	0.00
	F1 score	0.00	0.48	0.70	0.00

the same tracing algorithm. We can treat this as a numerical optimization problem. For any tunable reconstruction program $\mathfrak{P}(I|\boldsymbol{\theta})$, in which image I and parameteres $\boldsymbol{\theta}$ are inputs and SWC model is the output, we define the optimization problem as

$$\min_{textbf\theta} \mathrm{E}(\mathcal{L}(\mathfrak{P}(I|\theta), T_g(I))|I) \tag{20}$$

where $\mathcal{L}$ is the loss function that can be computed from reconstruction metrics.

In real applications, we expect parameters optimized on a training dataset can be generalized to other images from the same imaging protocol. Therefore, we carried out a cross-validation experiment on four image blocks (**Figure 5**) from a whole mouse brain sample acquired by fMOST (Gong et al., 2016). The cross-validation searched for the best parameters for each block and used these optimized parameters to trace other blocks. In our experiment, we used the F1 score of the SSD metric as the loss function to optimize the automatic neuron tracing method used in neuTube (Zhao et al., 2011), which has two numerical parameters for adjusting the sensitivity of branch detection. The optimization process was performed by simulated annealing (Van Laarhoven and Aarts, 1987), which searches the parameters iteratively. A new parameter $\theta^{(k+1)}$ at the kth iteration was calculated by

$$\theta^{(k+1)} = \theta^{(k)} + 20\frac{u}{|u|} * t_k * ((1 + \frac{1}{t_k})^{|u|} - 1) \tag{21}$$

where u was drawn randomly from $[-1, 1]\backslash 0$ and t_k was the temperature at the kth iteration. Starting from $t_1 = 0.01$, the temperature was decreased every 25 iterations at the rate of 0.96. The stop criterion was that the temperature was below 10^{-5} or the optimal value had not been improved for 20 iterations.

The results show that the optimized parameters outperformed the default parameters consistently, no matter which image block was used in parameter searching (**Figure 6**), presenting a successful example of using PyNeval in improving automatic neuron tracing.

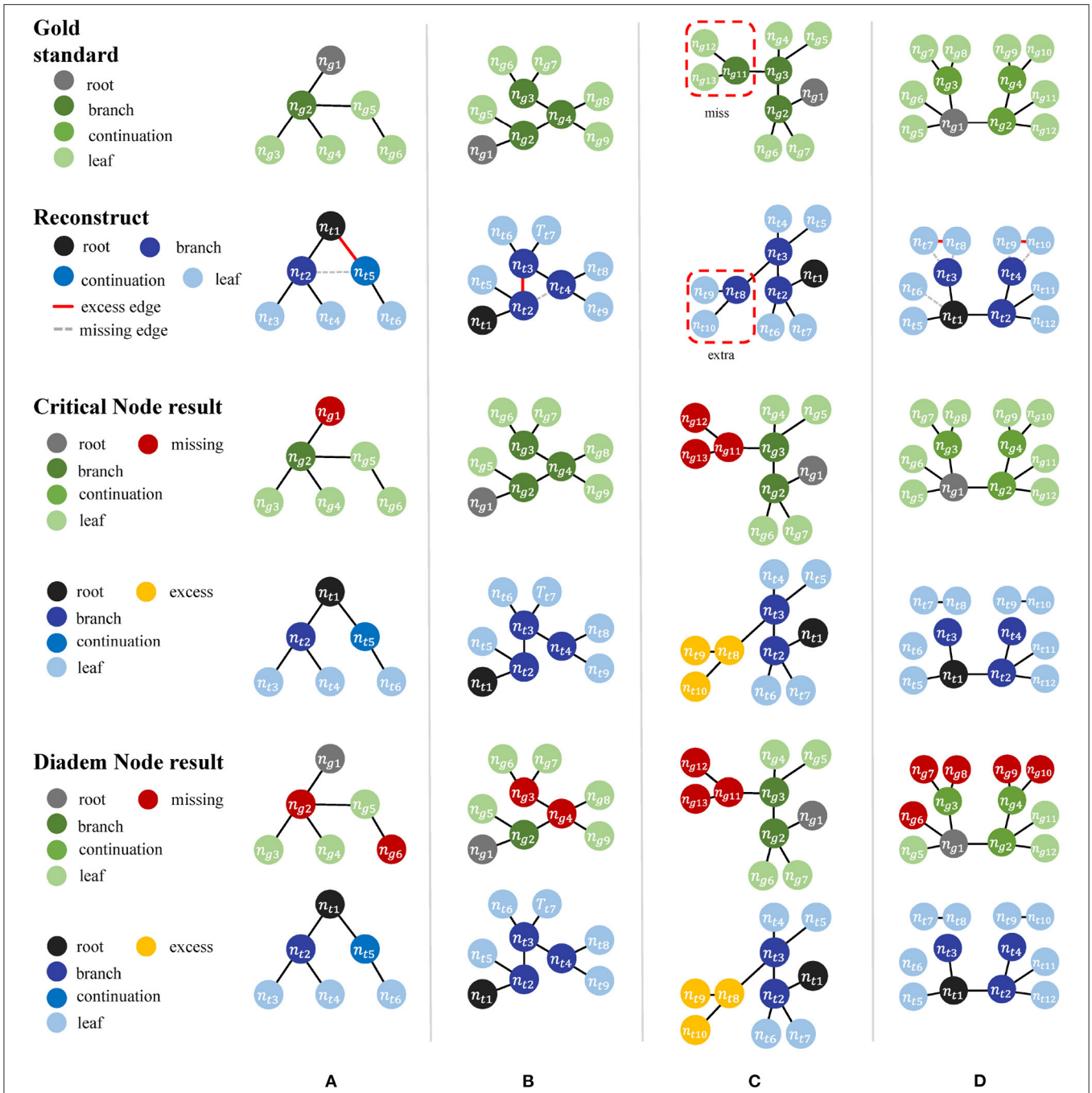


FIGURE 4 | (A–D) Four manually constructed cases for testing topological metrics.

TABLE 4 | PyNeval results of the DIADEM and length metrics for the topological cases are shown in **Figures 4A–D**.

Method	Index	file name			
		A	B	C	D
Diadem metric	Score	0.625	0.56	0.69	0.72
Critical node metric	Recall	0.80	1.00	0.70	1.00
	Precision	1.00	1.00	0.70	1.00
	F1 score	0.89	1.00	0.70	1.00

4. CONCLUSION AND FUTURE WORK

Motivated by the difficulties of evaluating automatic neuron tracing methods, we have developed PyNeval, a user-friendly Python toolbox to help method developers focus on algorithm development and method users choose a proper method for their own applications. PyNeval has made four popular metrics that cover both the geometrical and topological categories easily accessible to the community. A user can easily install PyNeval through common Python package managers and run the

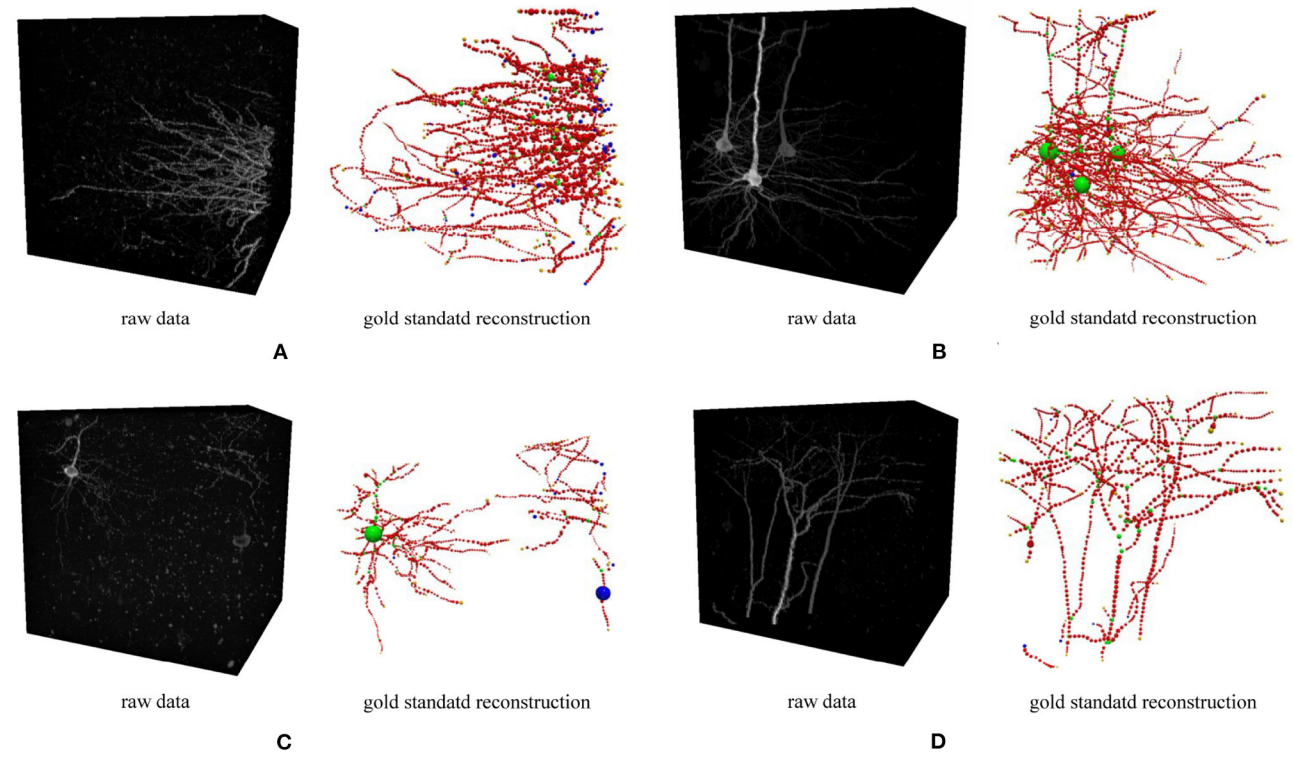


FIGURE 5 | Four 3D neuron images used in the optimization experiment. Each image and its gold standard reconstruction is rendered side by side in each panel labeled by the corresponding dataset ID. **(A)** FM3, **(B)** FM4, **(C)** FM5, and **(D)** FM6.

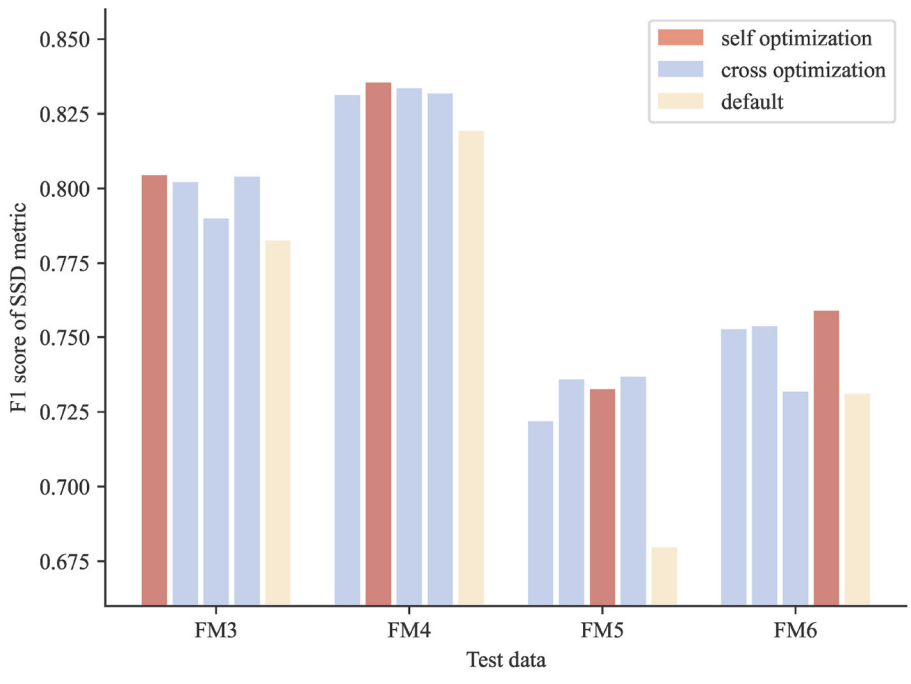


FIGURE 6 | Cross-validation results of parameter optimization for neuron reconstruction. The scores of the optimized parameters are consistently better than those of the default parameters for all the test images.

program as a command line with a straightforward but flexible interface. We have also shared the source code of PyNeval on https://github.com/CSDLLab/PyNeval to show how the metrics were implemented exactly as well as inspire further development.

To facilitate further development, PyNeval has a well-modularized architecture for maximizing its extensibility. It is straightforward to add more metrics such as the NetMets metric (Mayerich et al., 2012) in the future while keeping backward compatibility. Another important plan for further development is to make PyNeval an easy-to-use Python library as well, so that, other users can easily call functions in PyNeval from Python code directly, or even contribute their own metrics to PyNeval.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The animal study was reviewed and approved by Zhejiang University.

AUTHOR CONTRIBUTIONS

TZ and NZ designed and supervised the project. HZ wrote most part of the software with help from YY and TZ. HZ, CL, and JD performed data analysis. HZ, TZ, and NZ wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This work is supported by the National Key R&D Program of China (2020YFB1313501), Zhejiang Provincial Natural Science Foundation (LR19F020005), National Natural Science Foundation of China (61972347, 61976089), and Hunan Provincial Science & Technology Project Foundation (2018RS3065, 2018TP1018).

ACKNOWLEDGMENTS

We thank Wenzhi Sun and Wei Wu for providing fMOST data.

REFERENCES

Acciai, L., Soda, P., and Iannello, G. (2016). Automated neuron tracing methods: an updated account. *Neuroinformatics* 14, 353–367. doi: 10.1007/s12021-016-9310-0

Anytree. (2020). https://pypi.org/project/anytree/ (accessed Auguest 31, 2021).

Bille, P. (2005). A survey on tree edit distance and related problems. *Theor. Comput. Sci.* 337, 217–239. doi: 10.1016/j.tcs.2004.12.030

Cannon, R. C., Turner, D. A., Pyapali, G. K., and Wheal, H. V. (1998). An on-line archive of reconstructed hippocampal neurons. *J. Neurosci Methods* 84, 49–54. doi: 10.1016/S0165-0270(98)00091-0

Feng, L., Zhao, T., and Kim, J. (2015). neutube 1.0: a new design for efficient neuron reconstruction software based on the swc format. *eNeuro* 2:ENEURO.0049-14.2014. doi: 10.1523/ENEURO.0049-14.2014

Gillette, T. A., Brown, K. M., and Ascoli, G. A. (2011a). The diadem metric: comparing multiple reconstructions of the same neuron. *Neuroinformatics* 9, 233–245. doi: 10.1007/s12021-011-9117-y

Gillette, T. A., Brown, K. M., Svoboda, K., Liu, Y., and Ascoli, G. A. (2011b). Diademchallenge. org: a compendium of resources fostering the continuous development of automated neuronal reconstruction. *Neuroinformatics* 9, 303–304. doi: 10.1007/s12021-011-9104-3

Gong, H., Xu, D., Yuan, J., Li, X., Guo, C., Peng, J., et al. (2016). High-throughput dual-colour precision imaging for brain-wide connectome with cytoarchitectonic landmarks at the cellular level. *Nat. Commun.* 7:12142. doi: 10.1038/ncomms12142

Halavi, M., Hamilton, K. A., Parekh, R., and Ascoli, G. (2012). Digital reconstructions of neuronal morphology: three decades of research trends. *Front. Neurosci.* 6:49. doi: 10.3389/fnins.2012.00049

KDtree. (2017). https://pypi.org/project/kdtree/ (accessed Auguest 31, 2021).

Mayerich, D., Bjornsson, C., Taylor, J., and Roysam, B. (2012). Netmets: software for quantifying and visualizing errors in biological network segmentation. *BMC Bioinformatics* 13, S7. doi: 10.1186/1471-2105-13-S8-S7

Oliphant, T. E. (2007). Python for scientific computing. *Comput. Sci. Eng.* 9, 10–20. doi: 10.1109/MCSE.2007.58

Parekh, R., and Ascoli, G. A. (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

Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E., et al. (2015). Bigneuron: large-scale 3d neuron reconstruction from optical microscopy images. *Neuron* 87, 252–256. doi: 10.1016/j.neuron.2015.06.036

Peng, H., Long, F., Zhao, T., and Myers, E. (2011). Proof-editing is the bottleneck of 3d neuron reconstruction: the problem and solutions. *Neuroinformatics* 9, 103–105. doi: 10.1007/s12021-010-9090-x

Peng, H., Ruan, Z., Long, F., Simpson, J. H., and Myers, E. W. (2010). V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.* 28, 348–353. doi: 10.1038/nbt.1612

Rtree. (2020). https://pypi.org/project/rtree/ (accessed Auguest 31, 2021).

Van Der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* 13, 22–30. doi: 10.1109/MCSE.2011.37

Van Laarhoven, P. J., and Aarts, E. H. (1987). "Simulated annealing," in *Simulated annealing: Theory and applications* (Berlin: Springer), 7–15.

Wang, Y., Narayanaswamy, A., Tsai, C.-L., and Roysam, B. (2011). A broadly applicable 3-d neuron tracing method based on open-curve snake. *Neuroinformatics* 9, 193–217. doi: 10.1007/s12021-011-9110-5

Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., et al. (2011). Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics* 9, 247–261. doi: 10.1007/s12021-011-9120-3

Conflict of Interest: 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Neuroinformatics

TECHNOLOGY AND CODE
published: 15 February 2022
doi: 10.3389/fninf.2022.828787

The Brain Observatory Storage Service and Database (BossDB): A Cloud-Native Approach for Petascale Neuroscience Discovery

Robert Hider Jr.†, Dean Kleissas†, Timothy Gion, Daniel Xenes, Jordan Matelsky, Derek Pryor, Luis Rodriguez, Erik C. Johnson, William Gray-Roncal† and Brock Wester†*

Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD, United States

Technological advances in imaging and data acquisition are leading to the development of petabyte-scale neuroscience image datasets. These large-scale volumetric datasets pose unique challenges since analyses often span the entire volume, requiring a unified platform to access it. In this paper, we describe the Brain Observatory Storage Service and Database (BossDB), a cloud-based solution for storing and accessing petascale image datasets. BossDB provides support for data ingest, storage, visualization, and sharing through a RESTful Application Programming Interface (API). A key feature is the scalable indexing of spatial data and automatic and manual annotations to facilitate data discovery. Our project is open source and can be easily and cost effectively used for a variety of modalities and applications, and has effectively worked with datasets over a petabyte in size.

Keywords: connectome, software, cloud, data, storage, imaging, electron microscopy, X-ray

OPEN ACCESS

Edited by:
William T. Katz, Janelia Research Campus, United States

Reviewed by:
Yongsoo Kim, Penn State Milton S. Hershey Medical Center, United States
Pat Gunn, Flatiron Institute, United States

***Correspondence:**
Brock Wester
brock.wester@jhuapl.edu

†These authors have contributed equally to this work

Received: *04 December 2021*
Accepted: *10 January 2022*
Published: *15 February 2022*

Citation:
Hider R Jr, Kleissas D, Gion T, Xenes D, Matelsky J, Pryor D, Rodriguez L, Johnson EC, Gray-Roncal W and Wester B (2022) The Brain Observatory Storage Service and Database (BossDB): A Cloud-Native Approach for Petascale Neuroscience Discovery. Front. Neuroinform. 16:828787. doi: 10.3389/fninf.2022.828787

1. INTRODUCTION

Mapping the brain to better understand cognitive processes and the biological basis for disease is a fundamental challenge of the BRAIN Initiative. Technological advances in neuroimaging have grown rapidly over the last ten years, making it almost routine to image high-resolution (sub-micron) brain volumes in many laboratories around the world using Electron Microscopy (EM) and X-Ray Microtomography (XRM), among other imaging modalities (Bock et al., 2011; Helmstaedter et al., 2013; Kasthuri et al., 2015; Lee et al., 2016; Dupre and Yuste, 2017; Witvliet et al., 2021). These datasets, which provide the means to resolve individual neurons and the individual connections (synapses) between them, are highly valuable for providing key insights into neural connectivity and neuroanatomical features. As these high resolution neuroimaging volumes grow in extent, however, substantial challenges have emerged, including efficient data storage, the computational and financial cost of indexing and querying, and the technical difficulty of big-data visualization (Helmstaedter et al., 2013; Lichtman et al., 2014).

As new tools for interrogating neuroimaging datasets at high resolutions advance and become more common, a centralized data-access and data-processing paradigm is needed in order to take advantage of economies of scale when operating at the tera- to petascale level. While research groups are beginning to embrace data archives, most treat the system as simply a place to

deposit finalized data, with raw datasets generated and stored in a custom format and analyzed and inspected with custom software. At increasing data scale, it is quickly becoming impossible for researchers to characterize many of the underlying properties. For many recently-generated image volumes approaching the petascale, it is likely that most of the dataset is never viewed in detail by a human. Additionally, conventional approaches for automatically or semi-automatically reconstructing neuronal maps focus on building methods for small volumes, and scaling these tools to operate on multi-terabyte or petabyte data volumes, is often significantly beyond the capabilities and budgets of a single research group.

Large datasets are incredibly rich in scientific content which should be shared with others to best leverage the investment of time and resources, and to fully exploit the value of the data. Due to the challenges in collection, storage, and analysis of terascale and petascale data volumes, few public datasets of this size are routinely shared, even though many such volumes exist on local, private storage, and many petabytes of new data are anticipated in the future from programs like the BRAIN Initiative and other future large scale programming (Mikula, 2016; Dorkenwald et al., 2019; Wilson et al., 2019; Morgan and Lichtman, 2020; Scheffer et al., 2020; Phelps et al., 2021; Witvliet et al., 2021).

We considered use cases such as the first fully-automated pipelines for processing and assessing XRM (Dyer et al., 2017) and EM datasets (Bock et al., 2011; Kasthuri et al., 2015; Lee et al., 2016) and work by many academic laboratories around the world to understand state-of-the-art approaches and their limitations. We emphasize that high-performance and scalable data storage is an essential component of any modern connectomics effort, due to the need for rapid, multi-user data access. In designing our Brain Observatory Storage Service and Database (BossDB), we researched several related efforts, including DVID[1] (Katz and Plaza, 2019) which excels in versioned terascale storage; CATMAID and Knossos (Saalfeld et al., 2009; Helmstaedter et al., 2011) which provide a mature manual annotation platform. We previously worked with NeuroData to develop ndstore (Burns et al., 2013), which originated and implemented many of the design principles necessary to store and access high-dimensional imaging datasets. These principles include (1) an efficient internal data representation and associated spatial indexing scheme; (2) an API to remotely access services; and (3) MATLAB and Python toolkits to facilitate usability. Based on this prior research and an understanding of the evolving requirements driven by new and maturing imaging modalities, we created a robust, cloud-native petascale datastore with a number of services and support tools (**Figure 1**).

2. METHODS

To enable large-scale, collaborative research we developed and deployed a cloud-native data archive to support the storage, analysis, and sharing of large spatial datasets. Service-oriented architectures have continued to grow in popularity and possess many appealing properties when designing a cloud-based data archive (Vogelstein et al., 2016). Our solution, BossDB, is deployed within the Amazon Web Services (AWS) ecosystem and has been robustly designed to leverage cloud capabilities and ensure a highly-available, scalable, and cost-efficient system. Other research teams have previously deployed their own instantiations of BossDB (Vogelstein et al., 2016; Dyer et al., 2017).

2.1. Spatial Database

The spatial database is the foundation of BossDB, and uses the strengths of the cloud to efficiently store and index massive multi-dimensional image and annotation datasets (i.e., multi-channel 3D image volumes). A core concept is our managed storage hierarchy, which automatically migrates data between affordable, durable object storage (i.e., Amazon Simple Storage Service or S3) and an in-memory data store (i.e., Redis), which operates as read and write cache database for faster IO performance with a tradeoff of higher cost. The BossDB cache manages a lookup index to determine the fastest way to return data to the user, taking advantage of data stored in the hierarchy. While this requires the use of provisioned (non-serverless) resources, this allows for storage of large volumes at a low cost, while providing low latency to commonly accessed regions. We utilize AWS Lambda to perform parallel IO operations between the object store layer and memory cache layer and DynamoDB for indexing. These serverless technologies allow BossDB to rapidly and automatically scale resources during periods of heavy operation without incurring additional costs while idle.

The BossDB spatial database is designed to store petascale, multi-dimensional image data (i.e., multi-channel three-dimensional image volumes, with optional time series support, **Figure 2**) and associated coregistered voxel annotations (**Figure 3**). In this context, voxel annotations are unsigned 64-bit integer (uint64) labels stored in a separate *channel* that is in the same coordinate frame as the source image data. Each unique uint64 value represents a unique *object* (e.g., neuron, synapse, organelle). A user can leverage annotations within various channels (e.g., "segmentation," "mitochondria") to create groups of voxels to define objects that have some semantic meaning, typically the result of manual annotation or automated processing. The database maintains an index of annotation locations, enabling efficient spatial querying and data retrieval (**Figure 4**).

The internal representation of volumetric data utilizes small cuboids, or 3D chunks of data (i.e., $512 \times 512 \times 16$ voxels, which can vary in dimension), which are stored in Amazon S3 as compressed C-order arrays. Cuboids are indexed using a Morton-order space-filling curve, which maps the 3D location of each cuboid to a single dimension. In addition, annotations are indexed so BossDB can quickly retrieve which annotation IDs exist in an individual cuboid, and in which cuboids a unique ID exists. With these indices, all of which are stored in auto-scaling Amazon DynamoDB tables, the BossDB API can provide spatial querying of annotations by ID and efficient retrieval of arbitrary data volumes. The database will also render and store a resolution hierarchy through downsampling of a dataset, which

[1]Distributed, Versioned, Image-Oriented Dataservice. Available online at: https://github.com/janelia-flyem/dvid (accessed October 10, 2017).

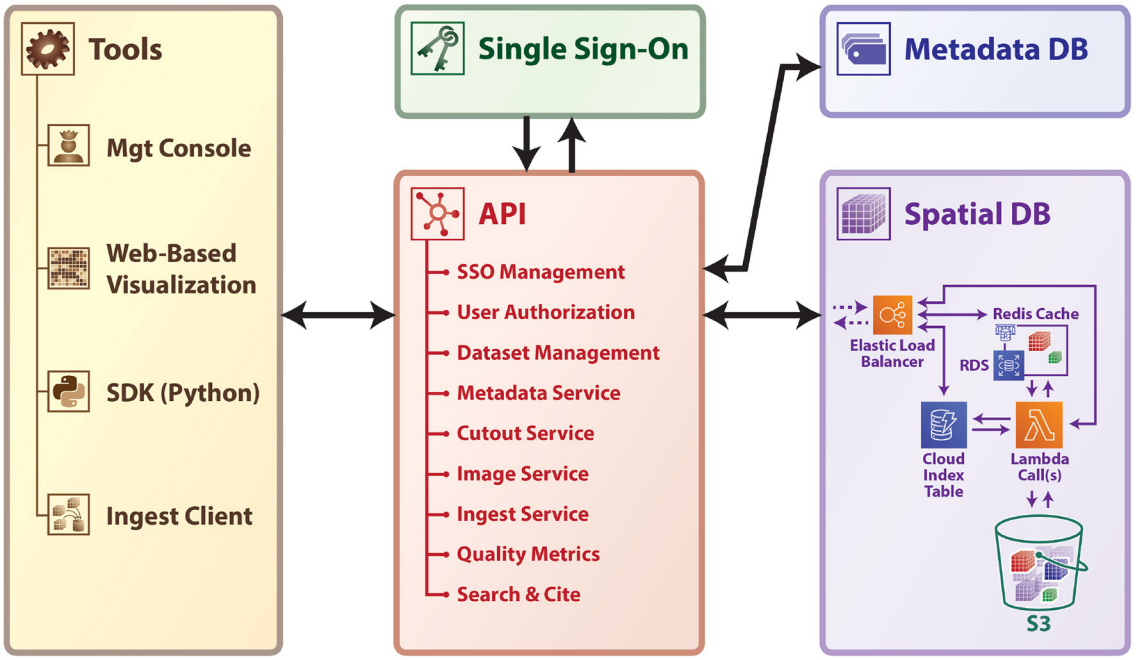


FIGURE 1 | A high-level schematic of BossDB platform.

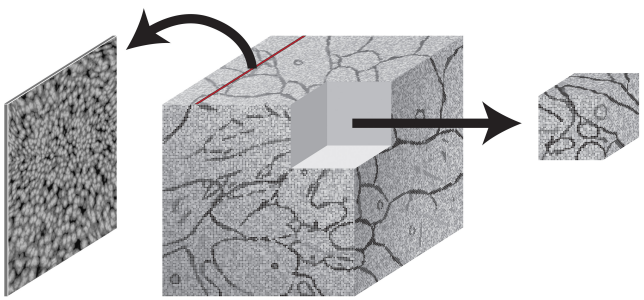


FIGURE 2 | An illustration showing image slices (left) being composed into 3D cuboid volumes (middle). Arbitrary requests may be made to extract image regions of interest (right).

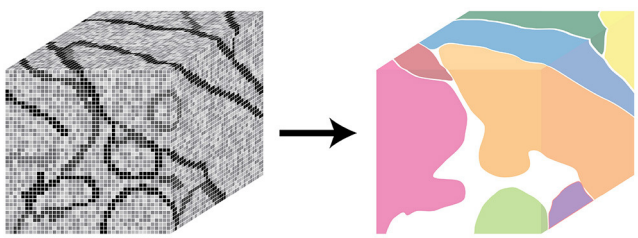


FIGURE 3 | An illustration showing *annotations*, composed of voxel labels (left) and how a unique annotation identifier can represent a unique *object* in the image data (right).

is critical for visualization applications to efficiently provide low-resolution views and useful when processing large datasets. The spatial database supports various bit-depths (including uint8, uint16 image channels and uint64 annotation channels) and we will provide additional bit-depth and data formats as needed.

Additionally, BossDB is able to store various mesh files associated with voxel annotation channel ID values, including precomputed format (Maitin-Shepard, 2021), which can be accessed through our API by visualization applications.

2.2. Single Sign-On Identity Provider

A centralized and standalone authentication server provides single sign-on functionality for BossDB and integrated tools and applications. This allows BossDB to control permissions internally and operate securely, while maintaining the ability to federate with other data archives in the future.

We use the open source software package Keycloak as an identity provider to manage users and roles. We created a Django OpenID Connect plugin to simplify the integration of services with the SSO provider.

Our identity provider server intentionally runs independently from the rest of BossDB system, forcing the BossDB API to authenticate just like any other SSO integrated tool or application, and making future federation with other data archives or authentication systems easy. The Keycloak server is deployed in an auto-scaling group that sits behind an Elastic Load Balancer in order to achieve high-throughput database requests with minimal latency.

2.3. Application Programming Interface

As the primary interface to BossDB, the API provides a collection of versioned, RESTful web services. It enforces access permissions and organizes data in a logical data model for spatial and functional results. Because the API is versioned,

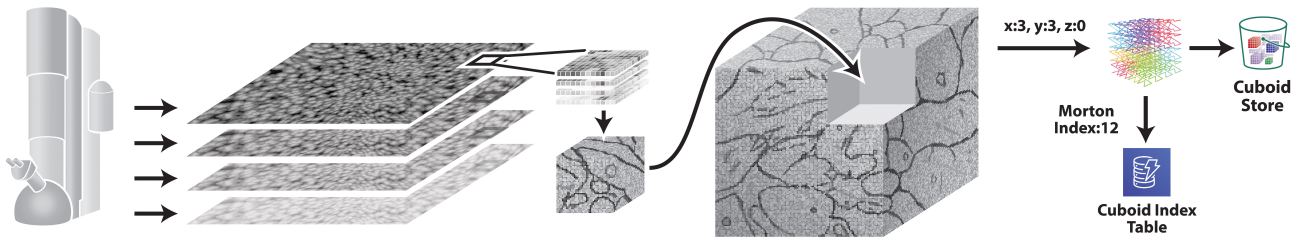


FIGURE 4 | An illustration showing how large 2D image slices generated by EM imaging are re-formatted as cuboids, which fit into a larger 3D volume, indexed using a *z*-order curve.

the BossDB storage engine can support significant changes while still maintaining backwards compatibility with legacy applications and tools. This BossDB API was designed from first principles to be versioned, and so this feature adds little in the way of day-to-day engineering complexity. All requests to the API are authenticated through the SSO service or via a long-lived API token, which enables tracking usage and throttling requests as needed to manage cost and ensure reliable performance (e.g., high bandwidth power user vs. a limited guest user). The services BossDB provides are summarized below:

2.3.1. SSO Management and User Authorization

A set of services to manage users, roles, groups, and permissions. Roles limit what actions a user can perform on the system, while permissions limit what data users can access or manipulate. Permissions are applied to BossDB datasets via groups, making it easy to manage and control access for both individuals and teams. Through the application of permissions, a researcher or administrator can choose to keep a dataset private, share with collaborators, or make it publicly available.

2.3.2. Dataset Management

The BossDB API organizes data into a logical hierarchy to group related data together (e.g., source image data and associated annotations, 2-photon and EM datasets from the same tissue sample). This service provides interfaces to create and manage datasets and their properties.

2.3.3. Ingest Service

A critical challenge when using a centralized data archive is the ingest of large datasets to standardized formats from diverse local storage formats and organization paradigms. The Ingest Service enables the moving of large datasets of varying data formats (**Table 1**) from local or cloud storage into BossDB by performing the upload of data into the cloud and then ingesting that data into the spatial database format, allowing independent scaling and failure recovery. The service provides methods to create a new ingest job, monitor the status of a job, join an upload client worker to a job, and cancel a job. Unlike general upload tools that run on client-side compute infrastructure, or commands like the *aws* command-line offerings that may run on a single host, the ingest client is able to perform ingests on arbitrarily many compute nodes, with graceful error management even in the case that a compute node powers down during an ingest job.

TABLE 1 | Data types and associated data formats that are supported by tile and chunk/volumetric based ingest service processes.

Data type	Data format	Ingest type
JPEG	8-bit, 16-bit	Tile
PNG	8-bit, 16-bit	Tile
TIFF	8-bit, 16-bit	Tile
CATMAID	Native format	Tile
HDF5	Any encoding	Tile/Chunk
N5	Any encoding	Tile/Chunk
Zarr	Any encoding	Tile/Chunk
CloudVolume	Native format	Tile/Chunk
DVID	Native format	Tile/Chunk
Nifti	Any encoding	Tile/Chunk
Dicom	Any encoding	Tile/Chunk
Knossos	Any encoding	Tile/Chunk

2.3.3.1. Tile Ingest

As demonstrated in **Figure 5**, the ingest process directly leverages AWS infrastructure, scaling on demand. First, using the ingest client a user uploads an ingest job configuration file to the API (1) which populates a task queue, enumerating all tiles that must be uploaded, and returns temporary AWS credentials. Next, the ingest client retrieves a task from the Upload Task Queue (2), and loads the requested local file into memory as an image tile (3), and uploads the tile data to an S3 bucket (4). The ingest client then writes a message to the index queue signaling it is finished with this tile (5). An AWS Lambda automatically fires when a message enters the Index Queue and it uses DynamoDB to track which tiles are successfully written to the tile bucket (6), (7) and when enough tiles in a region have arrived to generate the BossDB cuboid data representation, a second Lambda function is triggered (8). This Ingest Lambda function then loads the specified tiles, reformats them into cuboids, inserts them into the Spatial DB S3 bucket, updates the Spatial DB cuboid index, and finally marks the temporary tiles for deletion (9). The ingest client supports both parallel and distributed operation, allowing users to maximize their network bandwidth, especially in the case where source data is organized into numerous small image files.

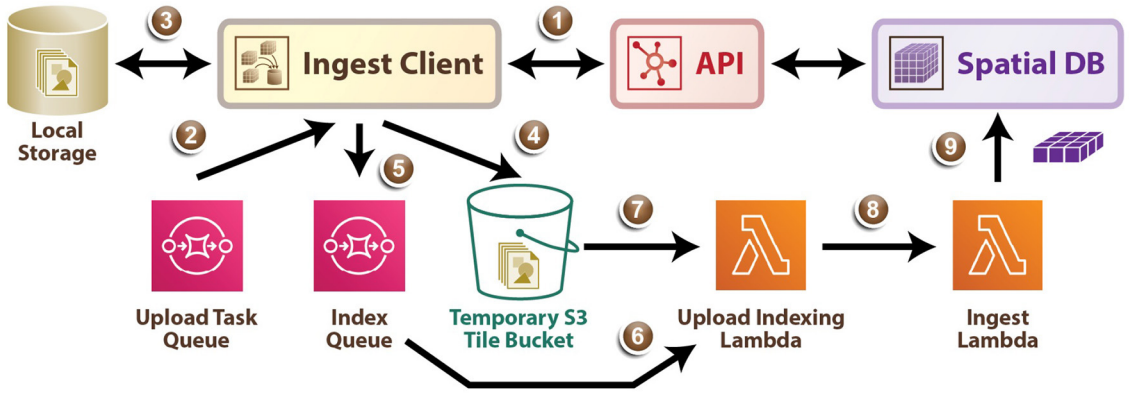


FIGURE 5 | A diagram outlining the data ingest service process.

2.3.3.2. Volumetric Ingest

The ingest process also supports uploading three-dimensional chunks of data in the CloudVolume format[2]; this interface can be straightforwardly extended to other formats. Similar to Tile Ingest, the ingest client is used to upload an ingest-job configuration file to the API, populating a task queue with all chunks to be uploaded. The ingest client then retrieves a task from the Upload Task Queue, and loads that chunk into memory. The memory chunk is divided into multiple BossDB cuboids (512 x 512 x 16) and each cuboid is uploaded to an AWS S3 bucket. Upon uploading, the S3 update will trigger an AWS Lambda that copies the cuboid into main s3 store, adds an entry in DynamoDB, and marks the original cuboid for deletion.

2.3.4. Dataset Metadata

BossDB can store arbitrary key-value pairs linked to data model items, which is useful to track experimental metadata and provenance (e.g., voxel size, animal information, annotation algorithm used). This service provides an interface to query, create, update, and delete key-value pairs associated with a dataset.

2.3.5. Cutout Service

BossDB provides the *cutout service*, which enables users to interact with the Spatial Database by reading and writing arbitrary data volumes. While BossDB stores all data internally using a standardized format, the cutout service uses HTTP content negotiation to determine the data format of a request, allowing users to request specific database-supported formats when downloading data (e.g., compressed C-order blob, hdf5 file, pickled numpy array). The same is true of data-uploads: A user-provided content annotation enables BossDB to accept data in a variety of volumetric and image-based formats. This service enables scalable analytics by letting users access arbitrary chunks of data in parallel, perform automated processing, and write the annotation result back to BossDB. It also supports querying for the spatial properties of annotations, such as the bounding box of an annotation or identifying which annotations exist within a region.

2.3.6. Image Service

In addition to our volumetric cutout service, we provide an image service to meet common user needs, which retrieves a 2D slice of data from the spatial database along one of the three orthogonal planes (i.e., XY, XZ, YZ), encoded as an image file. Again, HTTP content negotiation is used to determine the format of the response (e.g., png, jpeg). The service supports arbitrary image sizes or a fixed tile size, which is often used by visualization tools.

2.3.7. Downsample Service

To allow users to quickly assess, process, and interact with their data, BossDB iteratively builds a resolution hierarchy for each dataset by downsampling the source data. This is a workflow that is run infrequently and on-demand, and needs to scale from gigabytes to petabytes of data. We developed a serverless architecture built on AWS Step Functions to manage failures and track process state. AWS Lambda is used to perform the underlying image processing in a parallel, scalable fashion. This approach helps to minimize costs since resources are only provisioned when needed and scale on-demand in a fully-automated paradigm. It is also possible to perform a partial downsample when only a portion of the original dataset has changed, saving the time and expense of re-running the process on the entire dataset. Image volumes with anisotropic native voxel sizes (e.g., x = 4 nm, y = 4 nm, z = 40 nm) are downsampled in the image plane dimensions (e.g., downsampling factors of x = 2, y = 2, z = 1) until block sizes reach near-isotropy (e.g. third downsample to resolution of x = 32 nm, y = 32 nm, z = 40 nm), after which they are downsampled equally in all dimensions. This remaining anisotropy diminishes higher in the downsampled hierarchy. In general, these levels are used primarily for visualization, and most analyses are performed at native or near-native resolutions (resolution 0 or 1).

[2]CloudVolume Is a Python Library for Reading and Writing Chunked Numpy Arrays From Neuroglancer Volumes in "precomputed" Format. Available online at: https://github.com/seung-lab/cloud-volume.

2.4. User Tools

User facing tools are required to make a data archive truly useful, easy to use, and well documented. We currently offer a web-based management console, an ingest client, and a client-side Python module called `intern` for programmatic interaction[3] (Matelsky et al., 2021). We have also integrated 3rd-party web-based data visualization tools. While BossDB API provides a rich interface to interact with the system, user friendly tools built on top of the API are important to increase utility and adoption by the community. We expect to mature and expand the scope of this tool library as community users build on the core BossDB technologies.

2.4.1. Web-Based Management Console

BossDB has a web interface that lets users perform common actions interactively in their browser (e.g., create a dataset, monitor an ingest job, share a dataset with a user). This Django-backed web application is the primary interface for most users and will expose much of the API's functionality through an intuitive graphical interface. From the console, a researcher is able to manage datasets, discover new data, and launch the visualization tool.

2.4.2. Web-Based Visualization

A critical capability to any data archive is the ability to easily visualize stored data. Whether inspecting ingested data, exploring a dataset, or sharing an interesting sample with a collaborator, the most common interaction with stored data will be through visualization. We integrated a version of Neuroglancer (Maitin-Shepard, 2021) to let users visually explore data stored in BossDB, and enable other visualization methods that provide abstraction over much of the API's complexity. The Neuroglancer interface may be used on all modern browsers and operating systems that support WebGL, including (as of the time of publication) Chrome version 51 or greater, Firefox version 46 or greater, and Safari 15.0 or greater. Through use of the imagery API, BossDB also supports mobile-friendly data visualization tools such as *Substrate* (Matelsky et al., 2020).

2.4.3. Immersive Visualization and Annotation

The BossDB volumetric API likewise supports 3D collaborative annotation through immersive virtual reality (VR) tools such as syGlass (Pidhorskyi et al., 2018), which can enable high-throughput annotation of large volumes of dense imagery. VR takes advantage of the natural parallax of stereoscopic vision, which can improve the visual perception of complex 3D structures.

2.4.4. Ingest Client

We have developed an open source ingest client in Python to manage uploading data to BossDB. The ingest process operates on a upload task queue which contains tasks specifying individual 2D tiles or 3D chunks of data to upload. To deal with the unique formats and file organization methods of diverse users, the client uses a simple plug-in design to import custom snippets of code responsible for taking a task, finding the right file, and loading the data into memory, which is then uploaded by the client. The work queue design allows copies of the client to be run distributed across compute nodes and in parallel on a single machine, substantially increasing throughput.

2.4.5. Python Software Development Kit (SDK)

To support developers and researchers who want to programmatically interact with BossDB, we developed a pip-installable Python library that provides abstraction over much of the complexity in the API. Data cutouts of arbitrary size can be efficiently retrieved from our archive, enabling easy integration with analytics tools. The current SDK, called `intern`, will continue to be expanded and supported to accommodate updates and additions to the existing BossDB system and user requests.

3. RESULTS

3.1. Motivating Application

Many of our design requirements for the BossDB ecosystem were motivated by the activities planned for the Intelligent Advanced Research Projects Activity (IARPA) Machine Intelligent from Cortical Networks (MICrONS) Program[4]. This effort seeks to enable the rapid advancement of artificial intelligence capabilities by creating novel machine learning algorithms that use neurally-inspired architectures and mathematical abstractions of the representations, transformations, and learning rules employed by the brain[4]. To guide the construction of these algorithms, the program centers around massive co-registered functional (e.g., two-photon calcium imaging) and structural (e.g., EM) neuroimaging experiments aimed at estimating the synapse-resolution connectome of a $1\,mm^3$ volume of mouse visual cortex, represented by nearly a petabyte of image and segmentation data, and using that information to constrain machine learning architectures. Our goal was to organize, store, and support the analysis of these large functional and anatomical datasets, and eventually enable public dissemination.

3.2. Deployment

We envision that this data archive will facilitate neuroscience inquiries through extensible, scalable processes, with a sample workflow outlined that includes data generation, data ingest, intra- and cross-dataset analysis, and multi-user data visualization in various workflows (e.g., data proofreading) outlined in **Figure 6**. During the IARPA MICrONS Program, a deployed instance of our BossDB system enabled concurrent proofreading operation by dozens of users, as well as the storage of a highly-available contiguous image volume that approached 2 PB of lossless EM image data (Bishop et al., 2021) using the *blosc* compression standard[5]. In addition to

[3]Intern Software Development Kit (sdk) Tools Page on Bossdb.org. Available online at: https://bossdb.org/tools/intern (accessed December 03, 2021).

[4]MICrONS: Machine Intelligence From Cortical Networks. Available online at: http://iarpa.gov/index.php/research-programs/microns (accessed October 31, 2017).

[5]Blosc Compressor. Available online at: http://blosc.org (accessed December 03, 2021).

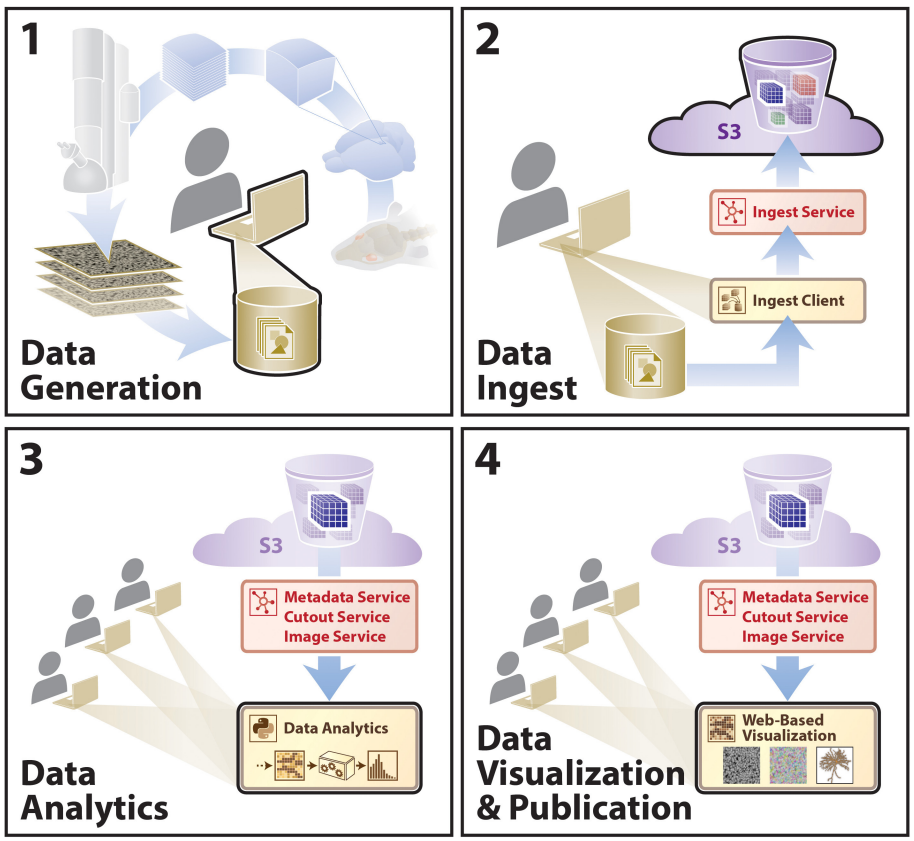


FIGURE 6 | A diagram outlining an example user story showing utilization of the BossDB infrastructure. A typical research group collecting data for a hypothesis will move sequentially from (1)–(4). Other groups will extend these analyses using steps (3) and (4). Sample data included for demonstration (see text footnote 4).

EM and segmentation datasets from the IARPA MICrONS program (https://bossdb.org/project/microns-minnie, https://bossdb.org/project/microns-pinky), we currently publicly store highly-available data for over 30 large-scale volumetric image collections, with multiple contiguous image volumes exceeding 100 TB in size (https://bossdb.org/projects/).

3.2.1. Implementation

Figure 7 shows the architecture of BossDB. The system has two user facing services: Authentication and Web Server Endpoint, both of which sit behind AWS elastic load balancers. The system uses Keycloak servers in a high-availability configuration for single sign-on authentication. The web server endpoints use Django API, to provide access to the majority of the services in BossDB.

BossDB uses serverless computing and storage, with AWS Lambda, SQS, S3, and DynamoDB to provide all of the other services mentioned in Section 2: Ingest, Metadata, Cutout, Image, and Downsample. Using serverless computing and storage for these components will automatically scale with demand and eliminate the need to maintain components.

BossDB is installed using the AWS CloudFormation service along with Salt and Packer to manage our infrastructure. This allows us to quickly duplicate the environment for testing and development and even change instance sizes within the new environments.

3.2.2. Data Generation

Researchers collect experimental data; stitching, alignment, and registration take part prior to upload to BossDB. Users create new resources in BossDB to identify and store their datasets, recording their experimental parameters and dataset properties (e.g., voxel dimensions, bit depth, spatial extent) prior to upload. An example screenshot from our web console is shown in **Figure 8**; this setup can be accomplished programmatically using `intern` as well.

3.2.3. Data Ingest

Once available, a researcher uploads image data via one of several methods supported by BossDB (e.g., REST API, ingest client), safely and efficiently storing data in BossDB. Large datasets can be uploaded incrementally, with data available for read as soon as it has been ingested, providing access to collaborators in minutes, not months.

The ingest client has already been used to upload petabytes of EM and calcium imaging data; many of these uploads proceed

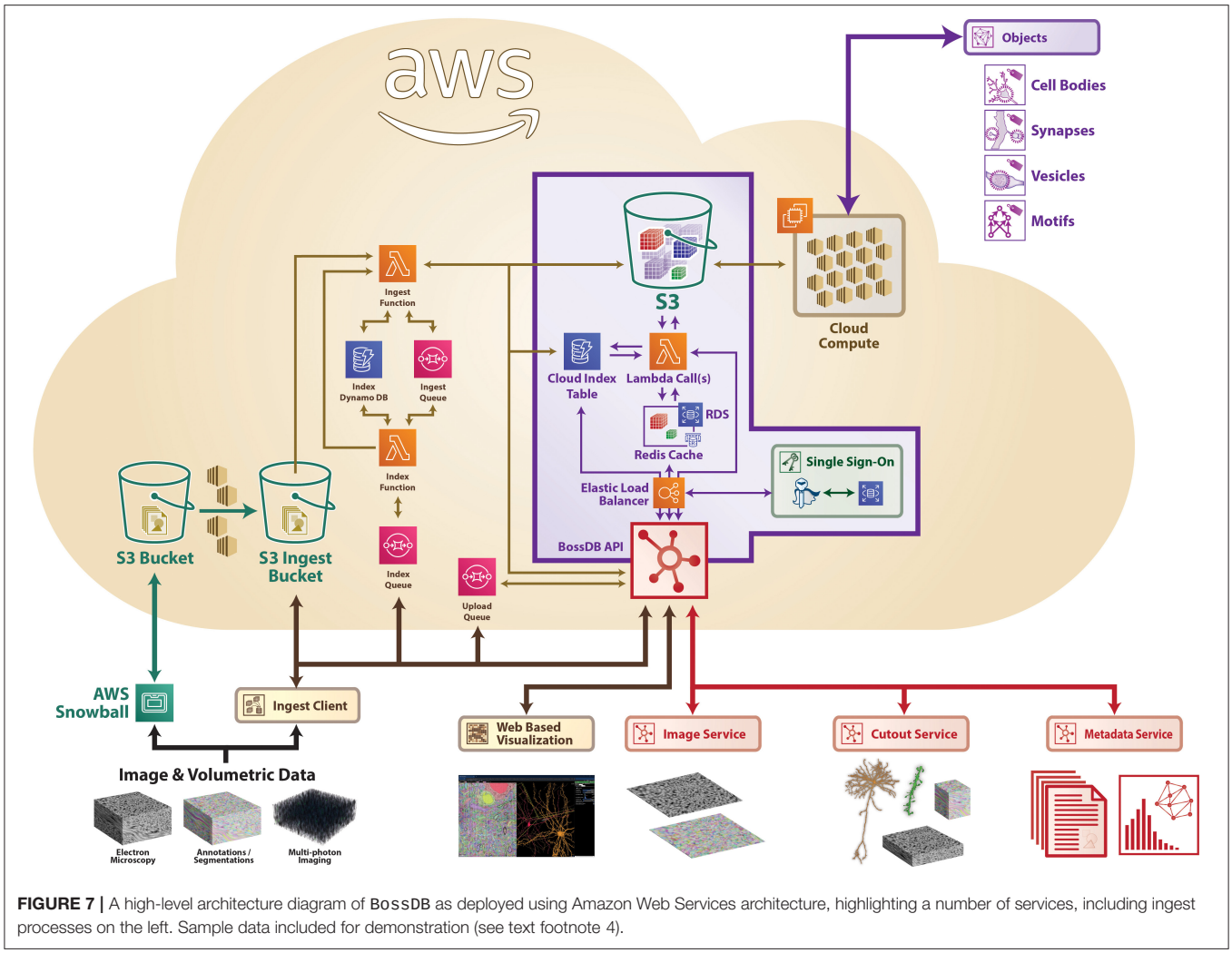


FIGURE 7 | A high-level architecture diagram of BossDB as deployed using Amazon Web Services architecture, highlighting a number of services, including ingest processes on the left. Sample data included for demonstration (see text footnote 4).

without any intervention from the developer team with the system automatically scaling to meet user's needs.

Previous testing of the ingest process reached a sustained ingest throughput of 230 GB/Min (**Figure 9**) using the volumetric ingest-client into BossDB. The ingest client was run on 750 kubernetes pods across eight large servers uploading data from an AWS Bucket. AWS Lambda scaled to over 5000 concurrent executing functions to handle the load.

To perform at this speed we were running 12 Endpoint servers sized with m4.2xlarge instances, an RDS database backed with a db.m4.xlarge instance, and DynamoDB table sized at 2,000 read / 4,000 write capacity.

This test shows the how BossDB will autoscale to meet demands (**Figure 10**). The same 3.2 million tiles from a 225-GB dataset were uploaded during each test. Each test used a different number of kubernetes pods running the ingest-client (100, 200, 400). BossDB automatically scaled endpoints, DynamoDB read and write demand to handle the throughput efficiently.

BossDB has monitoring capability at several levels. In **Figure 11** you see a snippet of our Ingest Dashboard which allows the administrator to see how much stress any one component of the system is under. Notifications will also go out if any key components fail, and when the system hits cost milestones.

3.2.4. Data Analytics

Many big data research analyses are enabled by BossDB features (e.g., standardized interfaces, arbitrary cutouts, spatial indexing), accelerating the scientific process.

One common use for BossDB is acting as a backend for local data analysis pipelines. Users download chunks of data from BossDB using `intern` and process it to create annotation labels using humans or machines. The resulting annotation data is uploaded via a choice of methods (python API, ingest client), below we include an example of such use case.

```
# import intern package
from intern import array

# specify data location
COLL_NAME = 'test_collection'
EXP_NAME = 'test_experiment'
```

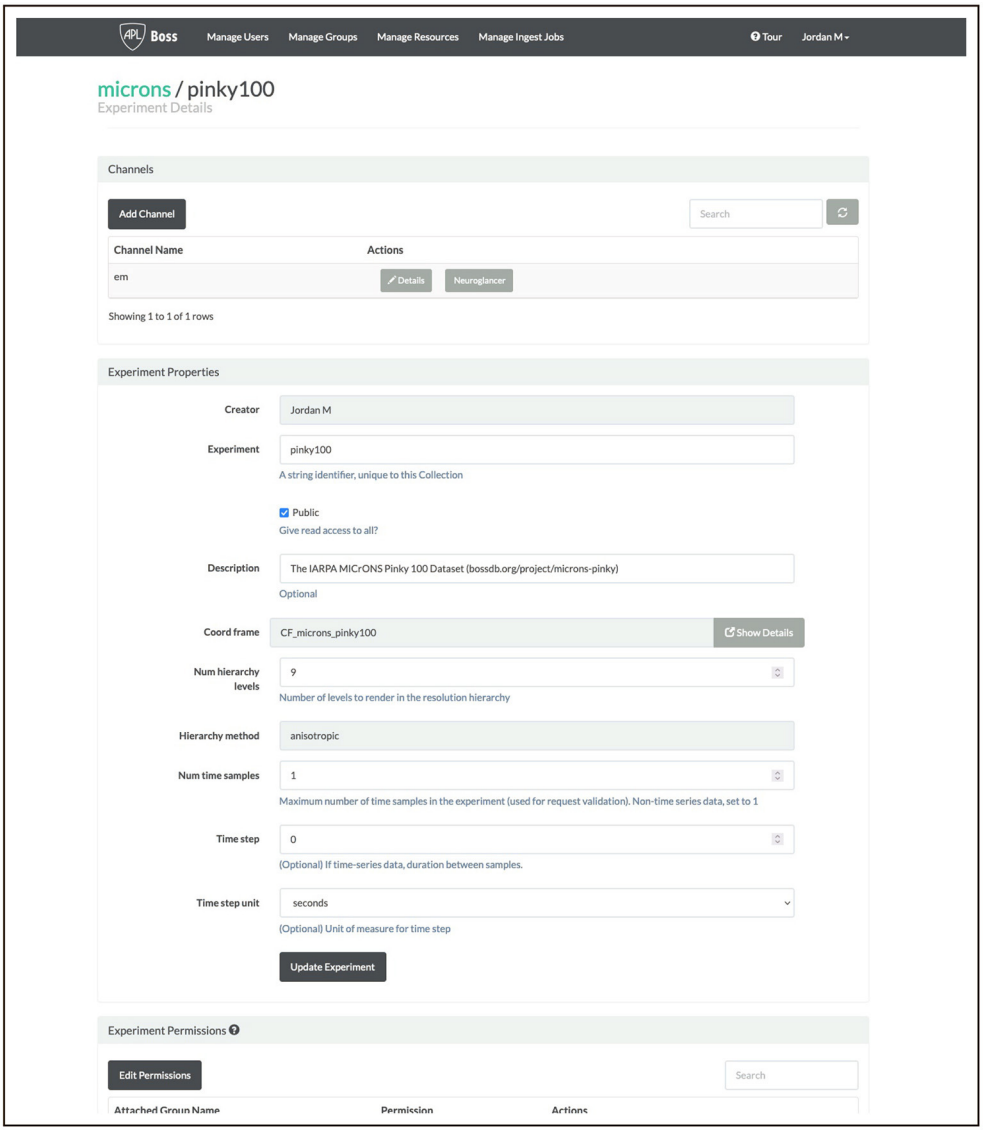


FIGURE 8 | An example screenshot from our BossDB console for the MICrONS Pinky dataset (see text footnote 4).

```
CHAN_NAME = 'test_channel'

# Use a URI to identify the data location:
chan = f"bossdb://{COLL_NAME}/{EXP_NAME}/
{CHAN_NAME}"

# Create a numpy-like pointer to the data,
# specifying the downsample-level:
dataset = array(chan, resolution=0)
# ...with access to dataset.shape,
  dataset.dtype, etc.

# Download the cutout from the channel into
  a 3D numpy array
data = dataset[0:10, 0:512, 0:1024].
```

3.2.5. Data Visualization and Publication

Data can be quickly visualized using applications such as Neuroglancer (**Figure 12**).

Data are published along with initial analysis, and made widely accessible through BossDB. Other research teams can then conduct additional analysis, extending and validating the existing scientific findings.

4. DISCUSSION

Our data archive will enable scientists to easily access and process large datasets, and to scale up their approaches with minimal alterations and without needing large local storage. Because the results are anchored to a universally-accessible datastore, it is

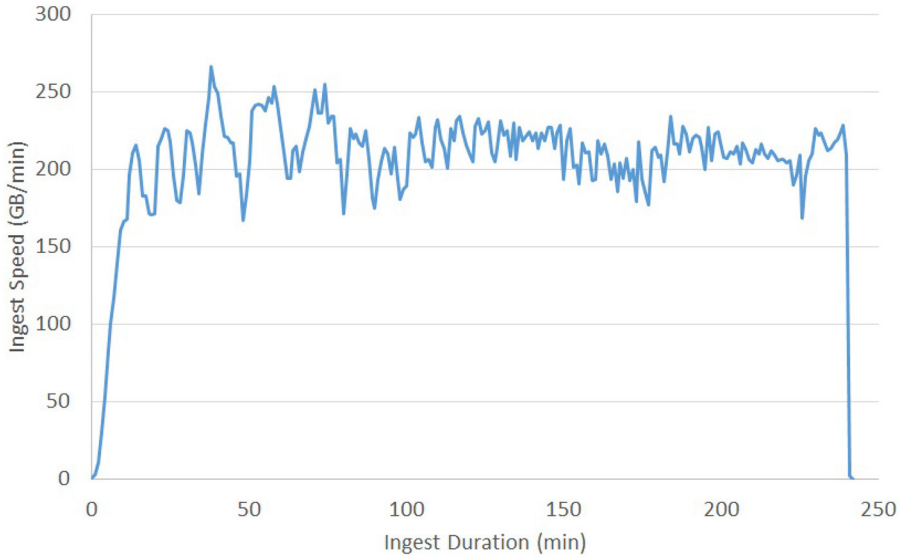


FIGURE 9 | Volumetric Ingest throughput demonstrated over the complete ingest of a 50TB dataset in about 4 h.

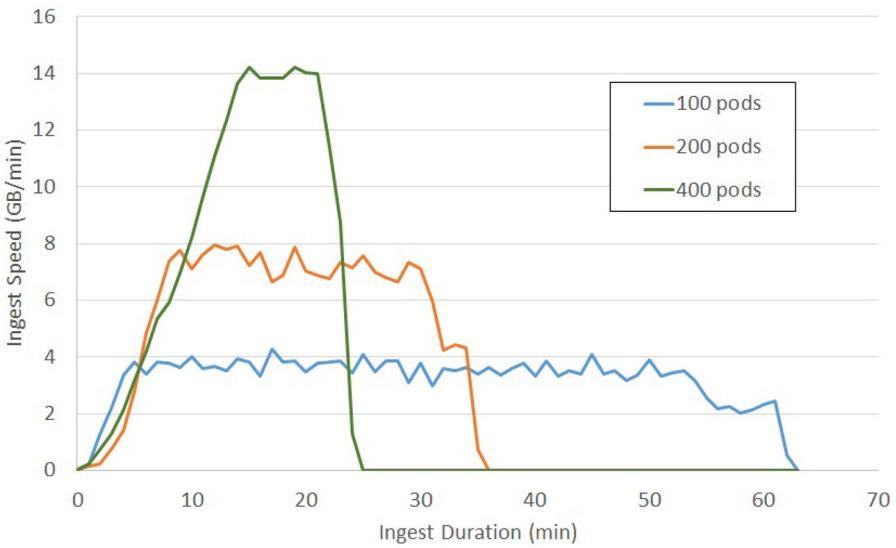


FIGURE 10 | Tile Ingest throughput on demand of a 200 GB EM dataset using various scales of ingest operation.

easier for others to inspect the results, improve upon them, and reproduce processing pipelines by leveraging common interfaces.

When considering a cloud-native approach, vendor lock-in is one potential concern – as we not only use the AWS cloud to deploy BossDB, but have integrated many of its services into the system to substantially accelerate development and performance. To minimize the development impact of expanding to an additional cloud provider or on-premise cluster, future work is needed to create a layer of abstraction between the core software and AWS services. We plan to continue to develop toward a microservices style architecture, which will decrease coupling between sub-components. This will allow BossDB to be able to independently scale sub-components and increase the ability to easily deploy, update, and manage services. We believe that storage engines will continue to specialize around datatypes (e.g., multi-dimensional image data, video data, gene sequence data) and be applicable to multiple research communities through the creation of domain-specific APIs that maintain the unique formats, organization, and needs of that community.

We intend to continue to provide BossDB as a reliable and scalable storage resource to the general microscopy and biology communities in perpetuity. We expect that as the

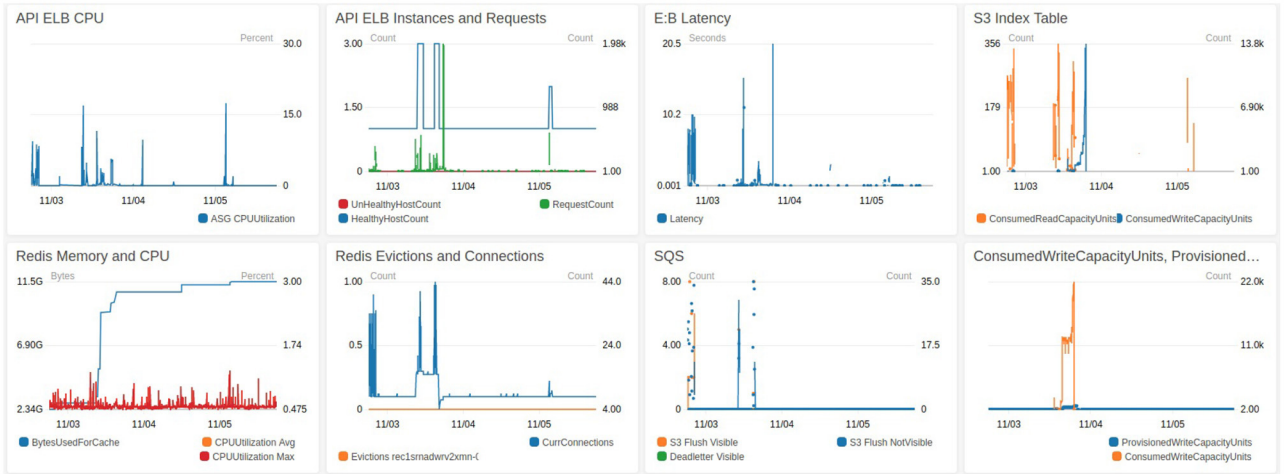


FIGURE 11 | A CloudWatch dashboard monitoring during ingestion.

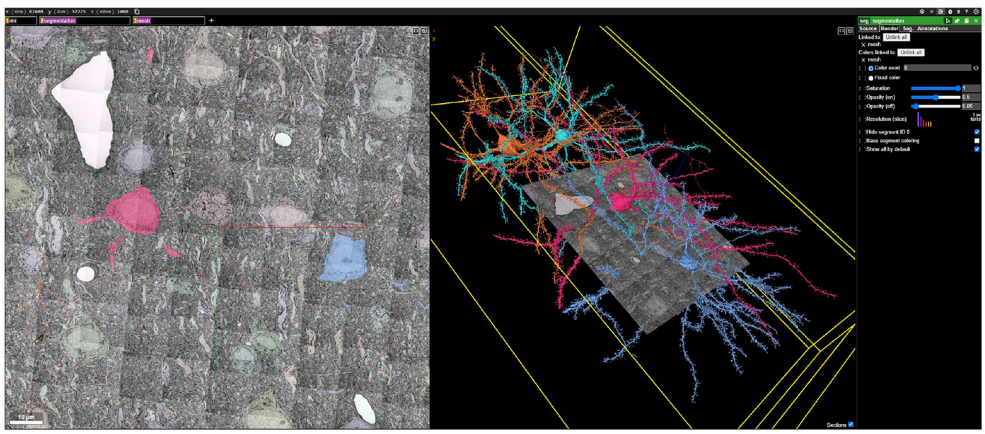


FIGURE 12 | An example Neuroglancer web visualization backended by BossDB, showing public EM, segmentation, and mesh data generated during the IARPA MICrONS Program (see text footnote 4).

community uses our data archive, additional tools will be developed to address new researcher needs, such as a universal, robust object-level metadata system and additional visualization engines. Several other research groups have leveraged BossDB deployments, including NeuroData (Vogelstein et al., 2018) which serves a diverse range of collaborators utilizing several imaging modalities (e.g., light microscopy, array tomography, serial multi-photon tomography) and added several new tools and capabilities to the BossDB ecosystem.

One concern about running a cloud data archive is estimating and managing cost. BossDB architecture was designed to allow dynamic scaling of resources to balance cost with performance and throughput capacity. As our software stack continues to mature, we plan to further optimize our tiered storage architecture (e.g., automatic migration data between S3 Standard, Infrequent Access, and Glacier tiers). The proposed system will provide a framework that is able to trivially scale from terabytes to petabytes while maintaining a balance between cost efficiency and performance.

As modern neuroscience datasets continue to grow in size, the community is fortunate to have several options to store and share their data. The precomputed format (Maitin-Shepard, 2021) offers a flexible, lightweight option that is readily deployable in both local and cloud settings. As mentioned above, DVID (see text footnote 1) is used to manage immutable and versioned annotations at the terascale level. We believe that our BossDB solution offers key advantages in scalability and indexing (adaptable from gigabyte to petabyte storage); authentication to manage user access workloads and costs; indexing to promote data exploration and discovery; and managed services to ensure that data is maintained and available in an efficient manner for a variety of user workflows. For a given research lab (or even within the lifecycle of a scientific question), one or more of these storage solutions may be most appropriate to enable and share results.

The standardization and scalability provided by our data archive will support a fundamental change in how researchers design and execute their experiments, and will rapidly accelerate the processing and reuse of high-quality neuroscience, most immediately for the large, petascale image, and annotation volumes produced by IARPA MICrONS. No previously existing platform met the operational and scaling requirements of the program, including managing an estimated 3–5 petabytes of image and annotation data—much larger than public neuroanatomical data archives. The BossDB software and documentation is open source and we are eager to expand the user community, supported modalities, and features. More information, examples and support are available at https://bossdb.org and https://github.com/jhuapl-boss/.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

RH, DK, TG, DX, JM, DP, LR, ECJ, WG-R, and BW contributed to the BossDB system design. RH, DK, TG, DX, JM, DP, and LR contributed to software development of the BossDB system. RH, DK, DX, WG-R, JM, and BW contributed to the manuscript drafting and reviews. All authors approved the submitted version of the manuscript.

FUNDING

This material is based upon work supported by the National Institutes of Health (NIH) grants R24MH114799, R24MH114785, and R01MH126684 under the NIH BRAIN Initiative Informatics Program and by the Office of the Director of National Intelligence (ODNI), and Intelligent Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2017-17032700004-005 under the MICrONS program.

ACKNOWLEDGMENTS

We would like to gratefully acknowledge our collaborators at NeuroData, including A. Baden, K. Lillaney, R. Burns, J. Vogelstein, B. Falk, and E. Perlman; S. Plaza and B. Katz for insights into DVID and volumetric frameworks; many contributors and facilitators, including J. Vogelstein, D. D'Angelo, C. Bishop, E. Johnson, H. Gooden, P. Manavalan, S. Farris, L. Kitchell, J. Downs, D. Ramsden, and D. Moore; and our user community.

REFERENCES

Bishop, C., Matelsky, J., Wilt, M., Downs, J., Rivlin, P., Plaza, S., et al. (2021). "CONFIRMS: a toolkit for scalable, black box connectome assessment and investigation," in *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE EMBC* (Mexico City).

Bock, D. D., Lee, W.-C. A., Kerlin, A. M., Andermann, M. L., Hood, G., Wetzel, A. W., et al. (2011). Network anatomy and *in vivo* physiology of visual cortical neurons. *Nature* 471, 177–182. doi: 10.1038/nature09802

Burns, R., Lillaney, K., Berger, D. R., Grosenick, L., Deisseroth, K., Reid, R. C., et al. (2013). "The open connectome project data cluster: scalable analysis and vision for high-throughput neuroscience," in *Proceedings of the 25th International Conference on Scientific and Statistical Database Management SSDBM* (Baltimore, MD: Association for Computing Machinery), 1–11.

Dorkenwald, S., Turner, N. L., Macrina, T., Lee, K., Lu, R., Wu, J., et al. (2019). *Binary and Analog Variation of Synapses Between Cortical Pyramidal Neurons.* Technical Report, Princeton University, Princeton, NJ.

Dupre, C., and Yuste, R. (2017). Non-overlapping neural networks in hydra vulgaris. *Curr. Biol.* 27, 1085–1097. doi: 10.1016/j.cub.2017.02.049

Dyer, E. L., Roncal, W. G., Prasad, J. A., Fernandes, H. L., Gürsoy, D., Andrade, V. D., et al. (2017). Quantifying mesoscale neuroanatomy using X-ray microtomography. *eNeuro* 4:ENEURO.0195-17.2017. doi: 10.1523/ENEURO.0195-17.2017

Helmstaedter, M., Briggman, K. L., and Denk, W. (2011). High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nat. Neurosci.* 14, 1081–1088. doi: 10.1038/nn.2868

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., and Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* 500, 168–174. doi: 10.1038/nature12346

Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., et al. (2015). Saturated reconstruction of a volume of neocortex. *Cell* 162, 648–661. doi: 10.1016/j.cell.2015.06.054

Katz, W. T., and Plaza, S. M. (2019). DVID: distributed versioned image-oriented dataservice. *Front. Neural Circ.* 13, 5. doi: 10.3389/fncir.2019.00005

Lee, W.-C. A., Bonin, V., Reed, M., Graham, B. J., Hood, G., Glattfelder, K., et al. (2016). Anatomy and function of an excitatory network in the visual cortex. *Nature* 532, 370–374. doi: 10.1038/nature17192

Lichtman, J. W., Pfister, H., and Shavit, N. (2014). The big data challenges of connectomics. *Nat. Neurosci.* 17, 1448–1454. doi: 10.1038/nn.3837.Epub

Maitin-Shepard, J. (2021). *Neuroglancer.* Available online at: https://github.com/google/neuroglancer (accessed June 10, 2017).

Matelsky, J., Rodriguez, L., Xenes, D., Gion, T., Hider, R. J., Wester, B., et al. (2021). "An Integrated toolkit for extensible and reproducible neuroscience," in *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE EMBC* (Mexico City), 2413–2418.

Matelsky, J. K., Downs, J., Cowley, H. P., Wester, B., and Gray-Roncal, W. (2020). A substrate for modular, extensible data-visualization. *Big Data Anal* 5, 1. doi: 10.1186/s41044-019-0043-6

Mikula, S. (2016). Progress towards mammalian whole-brain cellular connectomics. *Front. Neuroanatomy* 10, 62. doi: 10.3389/fnana.2016.00062

Morgan, J. L., and Lichtman, J. W. (2020). An individual interneuron participates in many kinds of inhibition and innervates much of the mouse visual thalamus. *Neuron* 106, 468–481.e2. doi: 10.1016/j.neuron.2020.02.001

Phelps, J. S., Hildebrand, D. G. C., Graham, B. J., Kuan, A. T., Thomas, L. A., Nguyen, T. M., et al. (2021). Reconstruction of motor control circuits in adult Drosophila using automated transmission electron microscopy. *Cell* 184, 759–774.e18. doi: 10.1016/j.cell.2020.12.013

Pidhorskyi, S., Morehead, M., Jones, Q., Spirou, G., and Doretto, G. (2018). syglass: interactive exploration of multidimensional images using virtual reality head-mounted displays. *arXiv [Preprint] arXiv:1804.08197*.

Saalfeld, S., Cardona, A., Hartenstein, V., and Tomancak, P. (2009). CATMAID: collaborative annotation toolkit for massive amounts of image data. *Bioinformatics* 25, 1984–1986. doi: 10.1093/bioinformatics/btp266

Scheffer, L. K., Xu, C. S., Januszewski, M., Lu, Z., Takemura, S.-Y., Hayworth, K. J., et al. (2020). A connectome and analysis of the adult drosophila central brain. *eLife* 9:e57443. doi: 10.7554/eLife.57443

Vogelstein, J. T., Mensh, B., Häusser, M., Spruston, N., Evans, A. C., Kording, K., et al. (2016). To the cloud! a grassroots proposal to accelerate brain science discovery. *Neuron* 92, 622–627. doi: 10.1016/j.neuron.2016.10.033

Vogelstein, J. T., Perlman, E., Falk, B., Baden, A., Gray Roncal, W., Chandrashekhar, V., et al. (2018). A community-developed open-source computational ecosystem for big neuro data. *Nat. Methods* 15, 846–847. doi: 10.1038/s41592-018-0181-1

Wilson, A. M., Schalek, R., Suissa-Peleg, A., Jones, T. R., Knowles-Barley, S., Pfister, H., et al. (2019). Developmental rewiring between cerebellar climbing fibers and purkinje cells begins with positive feedback synapse addition. *Cell Rep.* 29, 2849–2861.e6. doi: 10.1016/j.celrep.2019.10.081

Witvliet, D., Mulcahy, B., Mitchell, J. K., Meirovitch, Y., Berger, D. R., Wu, Y., et al. (2021). Connectomes across development reveal principles of brain maturation. *Nature* 596, 257–261. doi: 10.1038/s41586-021-03778-8

Author Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NIH, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation therein.

Conflict of Interest: 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Neuroinformatics

METHODS
published: 02 March 2022
doi: 10.3389/fninf.2022.828169

RealNeuralNetworks.jl: An Integrated Julia Package for Skeletonization, Morphological Analysis, and Synaptic Connectivity Analysis of Terabyte-Scale 3D Neural Segmentations

Jingpeng Wu[1]†, Nicholas Turner[1,2], J. Alexander Bae[1,3], Ashwin Vishwanathan[1] and H. Sebastian Seung[1,2]*

[1] Princeton Neuroscience Institute, Princeton University, Princeton, NJ, United States, [2] Department of Computer Science, Princeton University, Princeton, NJ, United States, [3] Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, United States

OPEN ACCESS

Edited by:
Ting Zhao,
Janelia Research Campus,
United States

Reviewed by:
Dawen Cai,
University of Michigan, United States
Daniel Raimund Berger,
Harvard University, United States

***Correspondence:**
Jingpeng Wu
jwu@flatironinstitute.org

†Present address:
Jingpeng Wu,
Center for Computational Neuroscience, Flatiron Institute, New York, NY, United States

Received: *03 December 2021*
Accepted: *10 February 2022*
Published: *02 March 2022*

Citation:
Wu J, Turner N, Bae JA, Vishwanathan A and Seung HS (2022) RealNeuralNetworks.jl: An Integrated Julia Package for Skeletonization, Morphological Analysis, and Synaptic Connectivity Analysis of Terabyte-Scale 3D Neural Segmentations. Front. Neuroinform. 16:828169. doi: 10.3389/fninf.2022.828169

Benefiting from the rapid development of electron microscopy imaging and deep learning technologies, an increasing number of brain image datasets with segmentation and synapse detection are published. Most of the automated segmentation methods label voxels rather than producing neuron skeletons directly. A further skeletonization step is necessary for quantitative morphological analysis. Currently, several tools are published for skeletonization as well as morphological and synaptic connectivity analysis using different computer languages and environments. Recently the Julia programming language, notable for elegant syntax and high performance, has gained rapid adoption in the scientific computing community. Here, we present a Julia package, called RealNeuralNetworks.jl, for efficient sparse skeletonization, morphological analysis, and synaptic connectivity analysis. Based on a large-scale Zebrafish segmentation dataset, we illustrate the software features by performing distributed skeletonization in Google Cloud, clustering the neurons using the NBLAST algorithm, combining morphological similarity and synaptic connectivity to study their relationship. We demonstrate that RealNeuralNetworks.jl is suitable for use in terabyte-scale electron microscopy image segmentation datasets.

Keywords: skeletonization, morphological analysis, clustering, connectomics, Julia language, neuron morphology, neuron connectivity

INTRODUCTION

Neural morphology and synaptic connectivity are closely related to brain function. With both nanometer resolution and a large field of view, advanced Electron Microscopes can produce large-scale image stacks (Kornfeld and Denk, 2018; Yin et al., 2020). Image voxels, pixels in a 3D image volume, can be clustered as individual neurons manually (Kasthuri et al., 2015) or automatically

using computer vision technologies (Lee et al., 2017, 2019, 2021; Januszewski et al., 2018; Macrina et al., 2021). Benefiting from the rapid development of deep learning (LeCun et al., 2015), the performance of automated segmentation approaches has greatly improved (Beier et al., 2017; Lee et al., 2017, 2019). With additional help from proofreading (Kim et al., 2014; Zhao et al., 2018; Dorkenwald et al., 2020; Hubbard et al., 2020), reconstructed neurons with synaptic connectivity can be used for scientific discovery (Deutsch et al., 2020; Januszewski et al., 2020; Vishwanathan et al., 2020).

Neurons are like trees and their skeletons can be used for morphological analysis. Skeleton or centerline representation is widely used in the morphological analysis (Stepanyants and Chklovskii, 2005; Halavi et al., 2008; Cuntz et al., 2011; Parekh and Ascoli, 2013; Armañanzas and Ascoli, 2015). In contrast to manual tracing and getting a neuron skeleton directly, most existing automated segmentation methods produce voxel labeling and are skeletonized in another step.

Synapses can also be detected automatically (Huang et al., 2018; Turner et al., 2020; Buhmann et al., 2021; Liu and Ji, 2021). Synaptic connectivity analysis can be used to detect motifs or communities. Although several software tools exist for each processing or analysis step, they were normally implemented using different computer languages. There is a lack of a consistent computational environment for the whole analysis pipeline, and users have to switch back and forth between different programming languages and environments.

Traditionally, developers normally use an interpreted language for prototyping, such as Python or MATLAB (MathWorks, Inc., Natick, MA, United States), and then translate the code to a compiled language, such as C or C++, to speed up the computation for large scale deployment. This was called a "two-language problem." Although some packages, such as Cython and pypy, can be used to help generate lower-level code, there still exist a lot of restrictions. Recently, a programming language with both intuitive syntax and high performance, called Julia (Bezanson et al., 2017), was designed to tackle this problem and has gotten more and more popular in the scientific computing community (Perkel, 2019). Benefiting from this design, prototype code can be compiled just in time and transformed into efficient binary code. As a result, we do not need to rewrite the prototype code using another low-level language, such as C or C++. Motivated by this elegant design, we use Julia to implement some essential analysis steps, including skeletonization, morphological analysis, and connectivity analysis, in two software packages called RealNeuralNetworks.jl and BigArrays.jl.

MATERIALS

We demonstrate the usage of RealNeuralNetworks.jl by analyzing a dataset with some proofread neurons. The details of this dataset, including sample preparation, imaging, automated segmentation, proofreading, was previously reported (Vishwanathan et al., 2017, 2021). Briefly, a sample (about 250 μm $\times$ 120 μm $\times$ 80 μm) from a zebrafish larvae brainstem was stained, sectioned, and imaged using a Zeiss Sigma field emitting scanning electron microscope. The image voxel size is 5 nm $\times$ 5 nm $\times$ 45 nm, and the final image volume size is over four terabytes with a voxel bit-depth of 8 (256 gray levels). Images are aligned and segmented automatically using a convolutional neural network (Lee et al., 2017; Wu et al., 2021). Based on the automated segmentation, about three thousand objects, including neurons or orphan neurites, were proofread using a modified Eyewire system (Kim et al., 2014; Greene et al., 2016; Bae et al., 2018; Vishwanathan et al., 2021). The final plain segmentation was exported to Google Cloud and visualized using Neuroglancer (Maitin-Shepard, 2021; **Figure 1**).

METHODS AND RESULTS

Data Storage

Segmentation and skeleton data are stored in Google Cloud Storage. The cutout and saving of segmentation chunks are implemented in a standalone Julia package, called BigArrays.jl (see section "Code Availability"). This is similar in functionality to the Python package CloudVolume (Charles et al., 2020; Silversmith et al., 2021b), and the data format is compatible with both packages. The cutout and saving of chunks were implemented on the client, so no intermediate server was needed. Benefiting from the distributed storage system in the cloud, the cutout and saving performance scales linearly with the number of operations. Besides skeletonization, BigArrays.jl was designed for general usage and could be used to handle arrays that are too large to fit in RAM. For example, a potential application is solving the out-of-memory issue in the simulation of quantum computing using tensor networks (Fishman et al., 2020) (Personal Communication).

For skeletonization, we can store the results in several formats. Currently, we support SWC with plain text encoding (Ascoli et al., 2001) which is widely used in most other analysis tools. Additionally, We also created a customized binary representation of SWC and all the numbers are encoded as binary scalar values directly and the loading and saving speed is greatly accelerated. For the synapses, it was detected externally and the result was saved using a language agnostic format "CSV."

Additionally, the data, including segmentation volume and skeletons, are formatted following Neuroglancer Precomputed. As a result, the data could be visualized directly using Neuroglancer (Maitin-Shepard, 2021) once they are uploaded to the cloud storage without any additional work.

Distributed Skeletonization of Neurons

To speed up skeletonization, we implemented the hybrid cloud distributed computation architecture in python-based chunkflow (Wu et al., 2021). The object IDs were used to define tasks and all the IDs were ingested to a queue in Amazon Simple Queue Service (SQS) using a Julia package called AWSSDK.jl (2021). The skeletonization of each neuron is independent of each other, so performance scales linearly with the number of nodes allocated.

Because task management (in SQS) and storage management are both distributed, we can launch workers on any computer with an internet connection and cloud authentication. Each

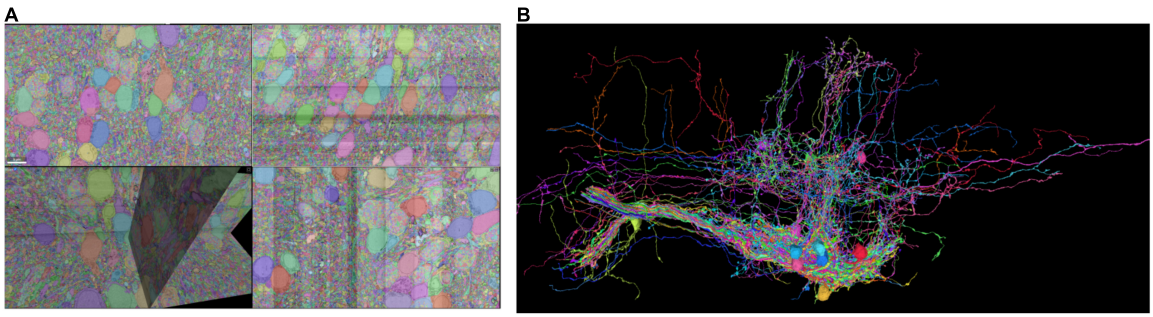


FIGURE 1 | Sparse segmentation after proofreading. **(A)** Some of the neurons are proofread and the fragments are agglomerated as individual neurons. **(B)** Some of the proofread neurons are visualized.

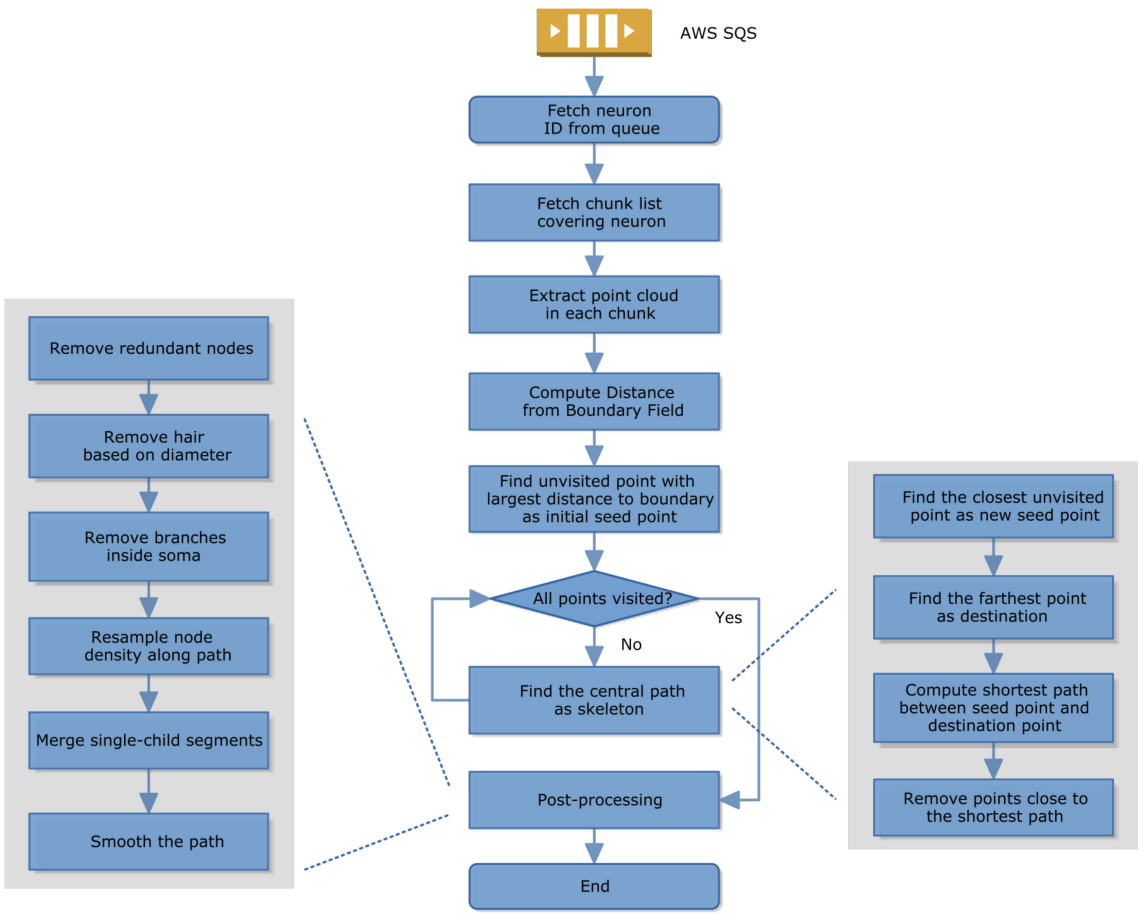


FIGURE 2 | Skeletonization computation in a worker.

task performs skeletonization for one object, called sparse skeletonization. The computation pipeline on the worker uses a modified TEASAR algorithm (Sato et al., 2000; Bae et al., 2018; Silversmith et al., 2021a). Briefly, the steps are as follows (**Figure 2**).

1. A worker fetches a task from SQS;
2. It then fetches the segmentation chunk list covering that object or neuron;
3. It extracts the point cloud of that object; It computes the distance from the boundary of the binary mask of that object;
4. It finds a point with the largest distance to the boundary as a seed;
5. If not all the points are visited, find a new central path by computing the shortest path from seed to the furthest unvisited point and then mark all the nearby points as unvisited;

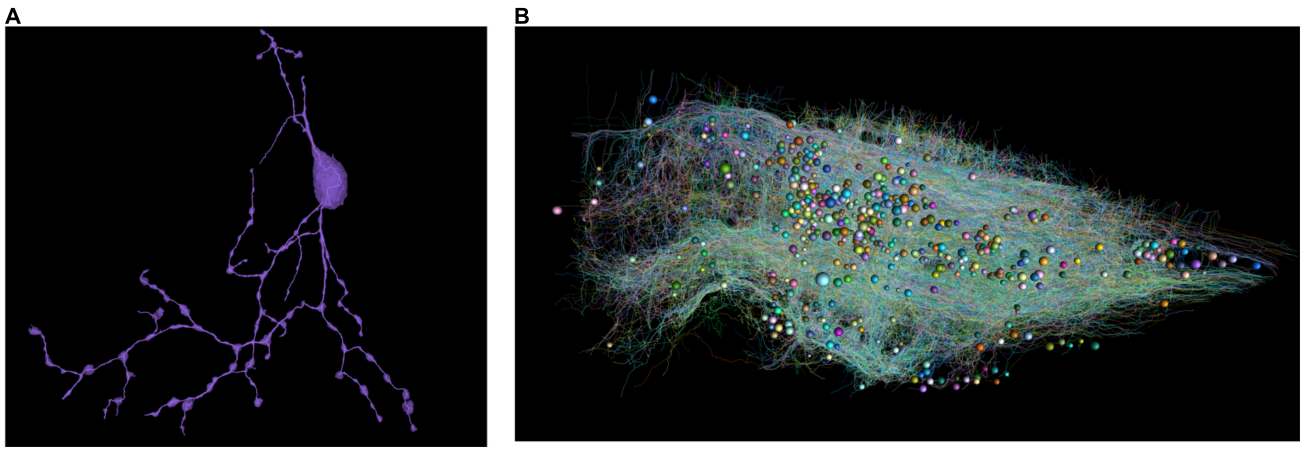


FIGURE 3 | (A) Skeletonize of a single neuron. Note that broken parts were reconnected. **(B)** All the skeletons with a random color. The spheres represent cell bodies with varying diameters.

6. If all the points have already been visited, the skeletonization is done and it switches to postprocessing, including removing redundant nodes, removing hair by comparing the diameter and path length, removing branches inside the cell body, resampling the node density to make it more evenly distributed along the path, removing empty branches, smoothing.

Given a sparsely or densely segmented volume, we extract the centerline or skeleton of its neurites one by one using a modified TEASAR algorithm (Sato et al., 2000; Bae et al., 2018; Silversmith et al., 2021a). Given a bit-packed binary volume representing a neuron, the foreground voxels are extracted as a point cloud. The distance from each point to the nearest boundary was computed as a Distance from Boundary Field (DBF). Find the point with the largest DBF as a seed point. Construct an undirected graph with points as nodes and neighboring points are connected with edges. Find the farthest unvisited point as the destination and compute the shortest path as the skeleton using Dijkstra's algorithm (Dijkstra, 1959). Points around the skeleton are marked as visited and not used in the following computation. Find the unvisited point closest to visited points as the new seed and iterate until all the points are visited. If the segmentation voxel is not continuous, we can look for the nearest terminal node (**Supplementary Figure 1**) to reconnect within a distance threshold. Note that the binary representation was bit-packed and the memory usage was reduced by 8 fold.

As a result, all proofread neurons are skeletonized (**Figure 3**). The distributed computation was performed in Google Cloud.

TABLE 1 | Features for single neuron morphology analysis.

Features	Description
Segment order	The order increases from the root node while branching
Segment length	The path length of a single segment
Branching angle	The angle of two segments in a branching point
Tortuosity	The curvature of a segment
Distance to root path length	The minimum path distance from the segment to root node
Average radius	The mean of all the nodes radius in the segment
Radius from soma	For each node, the Euclidean distance from the soma
Terminal segment path length	The path length of each terminal segment
The ratio of neck diameter to head	Could be used to identify spines

Morphological Features for Single Neuron Analysis

We decompose each neuron into segments or single nodes and compute their features. Definitions of node, branching node, root node, terminal node, segment, and the terminal segment are in **Supplementary Figure 1**. Additionally, an irreducible node corresponds to a soma, branching node, or terminal node. Based on existing literature (Uylings and van Pelt, 2002; Schierwagen, 2008; Schierwagen et al., 2010; Cuntz et al., 2011), we implemented some widely used morphological features for the skeletons and demonstrated the results using our zebrafish dataset (**Table 1** and **Figure 4**). In the spines of mammalian brains, the diameter of the neck is normally much smaller than the head, thus we added a feature to measure the ratio of neck diameter to head (**Figure 4H**).

Morphological Features of Many Neurons

For a number of neurons, we would like to encode each neuron using a feature vector, which could be used in neuron type clustering. Based on the literature, we have also implemented several widely used features (**Table 2**) and applied them to our zebrafish dataset (**Supplementary Figure 2**; Uylings and van Pelt, 2002; Schierwagen, 2008; Schierwagen et al., 2010; Cuntz et al., 2011; Wanner et al., 2016).

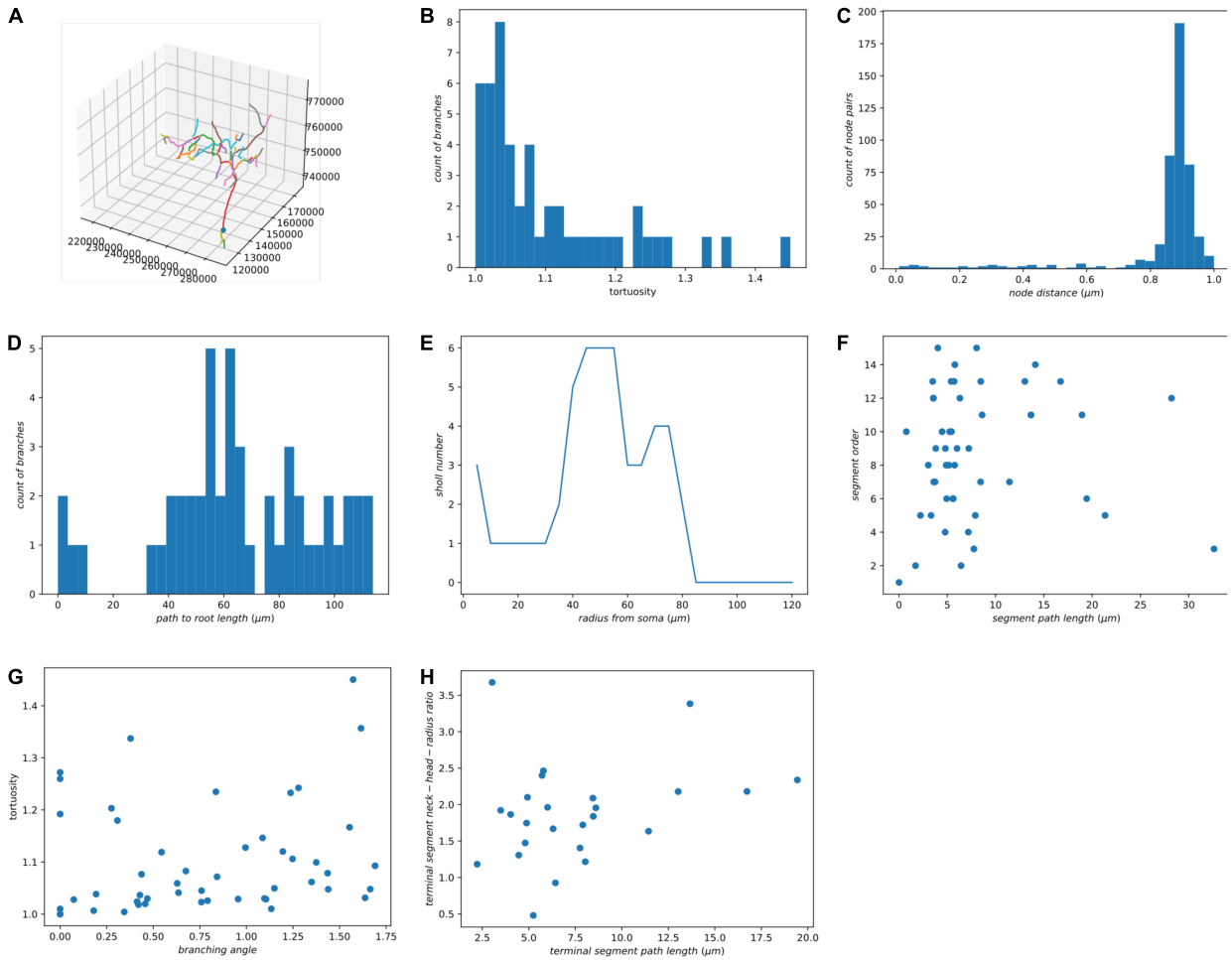


FIGURE 4 | Some morphological features of a single neuron. **(A)** The morphology of a neuron is visualized in Jupyter Notebook. **(B)** Histogram of tortuosity of neuron segments. **(C)** Histogram of neighboring node distance. **(D)** Histogram of path length to the root node. **(E)** Sholl analysis. **(F)** Segment path length versus segment order. **(G)** Branching angle in radians versus tortuosity of segments. **(H)** Terminal segment path length versus terminal segment neck-head radius ratio.

Morphological Clustering Using NBLAST

Most of traditional morphological features do not measure the spatial distribution of neurons. An automatic neuron type classification method, called NBLAST (Costa et al., 2016), measures the spatial distribution and is getting popular. The original method was implemented in R and C++. In order to incorporate this method in our analysis ecosystem, we implemented this algorithm from scratch using Julia. We performed hierarchical clustering (**Supplementary Figure 3**) using Clustering.jl (Stukalov and Lin, 2021) and classified the neurons into 23 types based on the NBLAST similarity scores (**Figure 5**). The visualization was created using Neuroglancer.

Synaptic Connectivity Combined With Morphology

After neuron segmentation and synapse detection were done externally, we can construct a graph of the neural network. Within the graph, the neurons are nodes and the synapses are edges. We use the synapse number as a connectedness metric for neurons. The more synapses connecting two neurons, the closer they are. Based on the distance matrix, we can perform hierarchical clustering, reorder the connectivity matrix, and identify some communities (**Figure 6A**).

Once we have the skeleton morphological features and synaptic connectivity, we can combine them. We can order the neurons in the connectivity matrix using NBLAST hierarchical clustering. As a result (**Figure 6B**), there are some morphologically similar neurons highly connected with each other. Morphologically similar neurons tend to have stronger synaptic connections as well (**Figure 6C**), which is consistent with previous findings in the mouse visual cortex (Lee et al., 2016).

DISCUSSION

RealNeuralNetworks.jl was built to process voxel segmentation datasets from Serial Section Electron Microscopy images.

TABLE 2 | Features of a single neuron.

Features	Description
Distance from soma to the center of skeleton mass	A metric to measure symmetricity centered by soma
Total path length	The physical length of all the skeleton paths
The number of branching points	
Median segment length	The median segment length of all the segments starts and ends at irreducible nodes
3D Sholl Analysis (Sholl, 1953)	Count the intersections to spheres centered on the root node
Average branching radian	The mean of the branching angles
Average tortuosity	The average value of the ratio of the path length to the Euclidean distance between irreducible nodes
Asymmetry	The distance of the soma node to the arbor center of mass
Typical radius	The Euclidian distance of the dendritic arbor points to the center of mass
Fractal dimension	Measures similarity across scales
Root node radius	The radius of the root node which is normally the soma
Total dendrite path length	If the dendrite segments are classified
Longest segment path length	
Convex hull volume	
Surface area	
Post-synapse number	Number of postsynaptic sites
Pre-synapse number	Number of presynaptic sites

Some components, such as skeletonization and morphological analysis, can be reused for sparsely labeled neurons in Light Microscopy images.

Comparison With Related Tools

Most existing tools are specifically designed for one or two analysis steps, rather than providing a one-stop solution and a consistent computational environment. Compared with some related software, RealNeuralNetworks.jl has a more complete toolset for the analysis (**Table 3**).

NeuTu (Zhao et al., 2018) was built mainly for proofreading neuron reconstruction from Electron Microscopy images. Besides that, it can also measure neuron shape similarity and perform clustering of neuron types (Zhao and Plaza, 2014). The measurement is built upon arbor density maps which is much more computationally heavy than skeleton-based NBLAST (Costa et al., 2016). Although the sparse skeletonization of NeuTu was also built upon the TEASAR algorithm (Sato et al., 2000), the geodesic distance between neighboring voxels is measured using the image intensity rather than distance map in our implementation. Thus, the skeleton accuracy is correlated with image quality.

Currently, RealNeuralNetworks.jl only has some widely used morphological features and is not as complete as L-Measure (Scorcioni et al., 2008) and TREES toolbox (Cuntz et al., 2010). Vaa3D (Peng et al., 2010, 2014) was built for light microscopy image processing, especially neuron tracing, and has a much richer set of tracing algorithms.

Kimimaro (Silversmith et al., 2021a) was built for dense skeletonization rather than sparse skeletonization. Currently, it does not have a bit-packed binary representation of segmentation volume and requires much more memory for sparse usage.

Why Julia

Julia is a modern language with nice features for both scientific computing and general programming (Bezanson et al., 2017; Perkel, 2019). It performs just-in-time compilation for the code, so performance can be comparable with C/C++. In addition, it has an intuitive syntax and an interactive programming interface like MATLAB (MathWorks, Inc., Natick, MA, United States), which is useful for prototyping and experiments. It is open-source with a permissive license, so it is much easier to deploy in the cloud compared with commercial languages requiring a license, such as MATLAB (MathWorks, Inc., Natick, MA, United States). Julia can be used interactively in Jupyter Notebooks (The "Ju" is from the name of Julia) (Perkel, 2019). Julia is increasingly popular in the scientific computing community. It has been downloaded over 25 million times and over 5000 packages are registered.

For most of the interpretable languages, such as Python and MATLAB, manipulating single elements in one or nesting loop is normally tens or hundreds of times slower than low-level languages, such as C and C++. For good performance, programmers are limited to using "vectorized" operations which were actually implemented in lower-level languages. In our applications, we perform a lot of voxel manipulations that are hard to express in "vectorized" operations. Benefiting from the just-in-time compilation, all of such operations can be implemented directly in Julia with good performance.

For the computation in local cluster or supercomputer, Julia was designed for distributed computing at the beginning and has gained a dramatic rise in the high-performance computing community. Our packages are expected to be adaptable in a local cluster.

Limitations

The skeletonization module was designed for sparse skeletonization rather than dense skeletonization. For sparse skeletonization, we can skeletonize some neurons of interest while the proofreading is ongoing. It would be too computationally expensive to iterate over the neurons individually in a terabyte-scale or petabyte-scale image volume. For dense skeletonization, Kimimaro is a better alternative (Silversmith et al., 2021a).

Currently, RealNeuralNetworks.jl only has limited support for visualization, such as functions for skeleton visualization. For more complicated plots, users must build their own scripts or Jupyter Notebooks based on other Julia visualization packages. Compared with the TREES toolbox (Cuntz et al., 2010, 2011), RealNeuralNetworks.jl does not have an interactive skeleton editing interface. Compared with

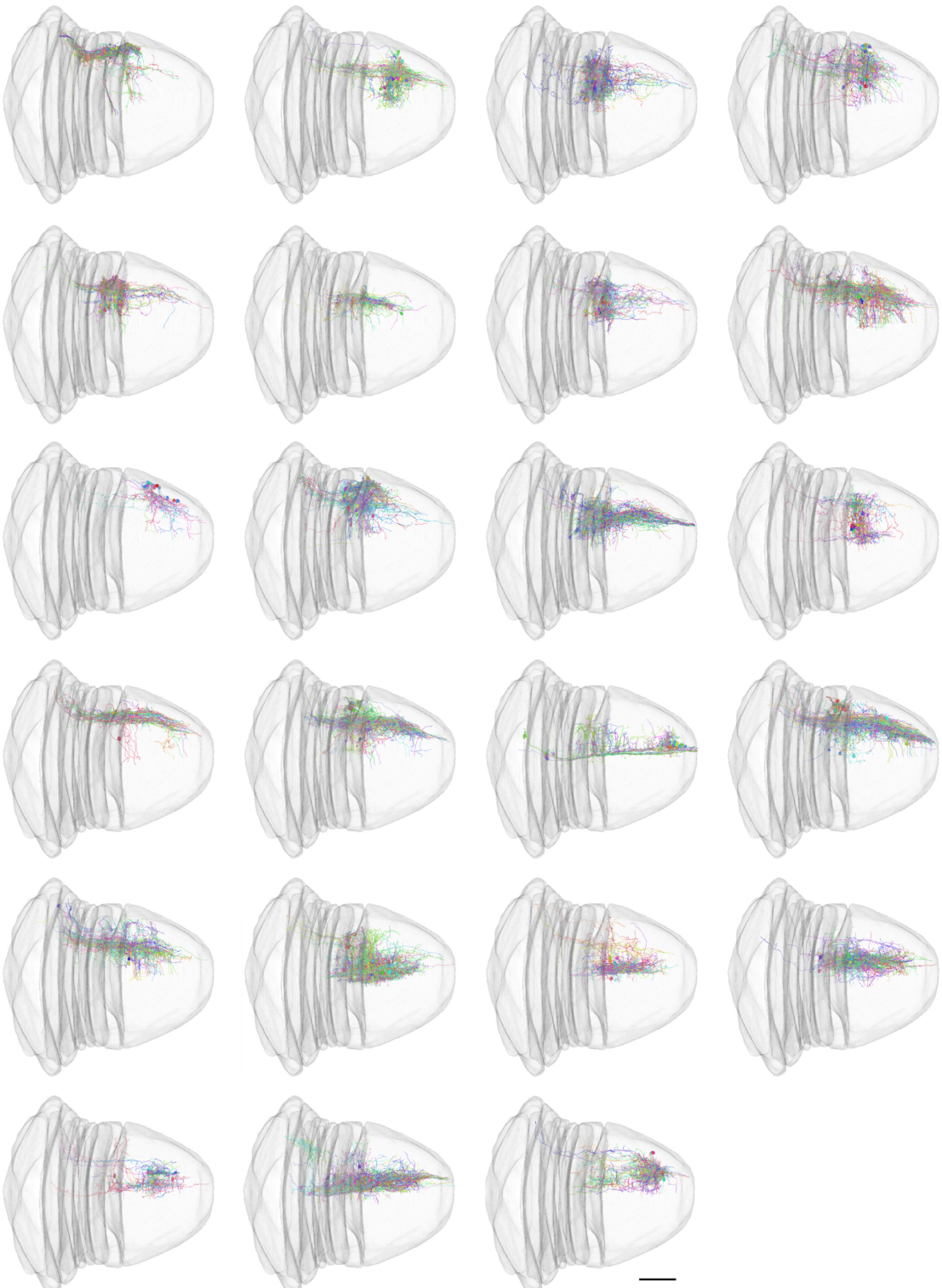


FIGURE 5 | NBLAST classification of neurons. The scale bar in the last image is 100 μm.

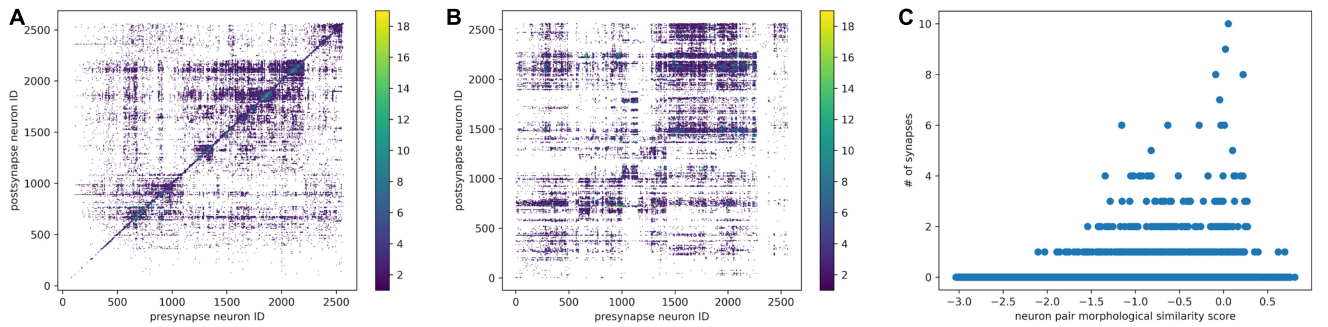


FIGURE 6 | Combine morphological NBLAST clustering and synaptic connectivity. **(A)** The synaptic connectivity matrix was reordered by hierarchical clustering based on connectivity distance. **(B)** The synaptic connectivity matrix was reordered according to hierarchical clustering based on the NBLAST score. The synapse number is encoded in the point diameter and color. **(C)** For each neuron pair, the relationship between NBLAST morphological similarity and number of synapses.

TABLE 3 | Comparison of software tools.

Tool/Feature	References	Language	Skeletonization	Morphological features	NBLAST similarity	Synaptic connectivity
L-Measure	Scorcioni et al., 2008	Java		✓		
NBLAST	Costa et al., 2016	R, C++			✓	
NeuroM	Palacios et al., 2021	Python		✓		
NeuTu	Zhao and Plaza, 2014	C++	✓			
TREES toolbox	Cuntz et al., 2010	MATLAB		✓		
Vaa3D	Peng et al., 2014	C++	✓	✓		
CBLAST	Januszewski et al., 2020	Python, R, C++			✓*	✓
3D BrainCV	Wu et al., 2014	MATLAB	✓	✓		
Kimimaro	Silversmith et al., 2021a	Python, C++	✓			
RealNeuralNetworks.jl		Julia	✓	✓	✓	✓

**CBLAST uses NBLAST for similarity measure.*

L-Measure, there are some missing morphological features in RealNeuralNetworks.jl.

Julia is a young language with rapid development and adoption in the scientific computing community. However, many of the packages are still evolving and are not yet stable.

CONCLUSION

In summary, we present a Julia-based tool, called RealNeuralNetworks.jl, for sparse skeletonization, morphological analysis, and synaptic connectivity analysis. We provide an integrated computational environment for the analysis pipeline. We have demonstrated the utility of this package by processing a Zebrafish segmentation dataset. We hope that it could be useful for other connectomics projects in the future.

CODE AVAILABILITY

The code is open-sourced in GitHub: https://github.com/seung-lab/RealNeuralNetworks.jl. The BigArrays.jl is available in GitHub as well: https://github.com/seung-lab/BigArrays.jl. The Jupyter Notebooks are available in GitHub: https://github.com/jingpengw/realneuralnetworks-notebook.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

JW implemented the software, performed the experiments, and wrote the manuscript. NT translated the MATLAB skeletonization code to Julia. JB improved the TEASAR algorithm and implemented it in MATLAB. AV contributed to sample preparation, imaging, and management of proofreading. HS designed and conceptualized the study. All authors contributed to the article and approved the submitted version.

FUNDING

HS acknowledges support from the NIH/NEI R01EY027036, NIH R01 NS104926, and R01 EY027036. HS acknowledges support from NIH/NCI UH2CA203710, ARO W911NF-14-1-0407, and Mathers Foundation, as well as assistance from Google, Amazon, and Intel. HS is grateful for support from the Intelligence Advanced Research Projects Activity (IARPA) via Department

of Interior/Interior Business Center (DoI/IBC) contract number D16PC0005. The United States Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright annotation thereon.

ACKNOWLEDGMENTS

We would like to thank Merlin Moore, Kyle Wille, Ryan Willie, Selden Koolman, Sarah Morejohn, Ben Silverman, Doug Bland, Celia David, Sujata Reddy, Anthony Pelegrino, Sarah Williams, and Dan Visser for manual annotation and validation, Amy Robinson for EyeWire management, Kisuk Lee for convolutional neural network training, Will Wong and William M. Silversmith for image data transformation for Eyewire, and William M. Silversmith and Pat Gunn for manuscript editing.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fninf.2022.828169/full#supplementary-material

Supplementary Figure 1 | Nomenclature of neuron skeleton parts.

Supplementary Figure 2 | Distribution of morphological features.

Supplementary Figure 3 | Hierarchical clustering using the NBLAST score.

REFERENCES

Armañanzas, R., and Ascoli, G. A. (2015). Towards the automatic classification of neurons. *Trends Neurosci.* 38, 307–318. doi: 10.1016/j.tins.2015.02.004

Ascoli, G. A., Krichmar, J. L., Nasuto, S. J., and Senft, S. L. (2001). Generation, description and storage of dendritic morphology data. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 356, 1131–1145. doi: 10.1098/rstb.2001.0905

AWSSDK.jl (2021). *JuliaCloud*. Available online at: https://github.com/JuliaCloud/AWSSDK.jl (accessed February 1, 2022).

Bae, J. A., Mu, S., Kim, J. S., Turner, N. L., Tartavull, I., Kemnitz, N., et al. (2018). Digital museum of retinal ganglion cells with dense anatomy and physiology. *Cell* 173, 1293.e19–1306.e19. doi: 10.1016/j.cell.2018.04.040

Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D. D., et al. (2017). Multicut brings automated neurite segmentation closer to human performance. *Nat. Methods* 14, 101–102. doi: 10.1038/nmeth.4151

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. (2017). Julia: a fresh approach to numerical computing. *SIAM Rev.* 59, 65–98. doi: 10.1137/141000671

Buhmann, J., Sheridan, A., Malin-Mayor, C., Schlegel, P., Gerhard, S., Kazimiers, T., et al. (2021). Automatic detection of synaptic partners in a whole-brain Drosophila electron microscopy data set. *Nat. Methods* 18, 771–774. doi: 10.1038/s41592-021-01183-7

Charles, A. S., Falk, B., Turner, N., Pereira, T. D., Tward, D., Pedigo, B. D., et al. (2020). Toward community-driven big open brain science: open big data and tools for structure, function, and genetics. *Annu. Rev. Neurosci.* 43, 441–464. doi: 10.1146/annurev-neuro-100119-110036

Costa, M., Manton, J. D., Ostrovsky, A. D., Prohaska, S., and Jefferis, G. S. X. E. (2016). NBLAST: rapid, sensitive comparison of neuronal structure and construction of neuron family databases. *Neuron* 91, 293–311. doi: 10.1016/j.neuron.2016.06.012

Cuntz, H., Forstner, F., Borst, A., and Häusser, M. (2010). One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS Comput. Biol.* 6:e1000877. doi: 10.1371/journal.pcbi.1000877

Cuntz, H., Forstner, F., Borst, A., and Häusser, M. (2011). The TREES Toolbox—Probing the basis of axonal and dendritic branching. *Neuroinformatics* 9, 91–96. doi: 10.1007/s12021-010-9093-7

Deutsch, D., Pacheco, D., Encarnacion-Rivera, L., Pereira, T., Fathy, R., Clemens, J., et al. (2020). The neural basis for a persistent internal state in Drosophila females. *eLife* 9:e59502. doi: 10.7554/eLife.59502

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numer. Math.* 1, 269–271. doi: 10.1007/bf01386390

Dorkenwald, S., McKellar, C., Macrina, T., Kemnitz, N., Lee, K., Lu, R., et al. (2020). FlyWire: online community for whole-brain connectomics. *bioRxiv [preprint]* doi: 10.1101/2020.08.30.274225

Fishman, M., White, S. R., and Stoudenmire, E. M. (2020). *The ITensor Software Library for Tensor Network Calculations. arXiv:2007.14822*. Available online at: http://arxiv.org/abs/2007.14822 (accessed November 2, 2021).

Greene, M. J., Kim, J. S., and Seung, H. S. (2016). Analogous convergence of sustained and transient inputs in parallel on and off pathways for retinal motion computation. *Cell Rep.* 14, 1892–1900. doi: 10.1016/j.celrep.2016.02.001

Halavi, M., Polavaram, S., Donohue, D. E., Hamilton, G., Hoyt, J., Smith, K. P., et al. (2008). NeuroMorpho.org implementation of digital neuroscience: dense coverage and integration with the NIF. *Neuroinformatics* 6, 241–252. doi: 10.1007/s12021-008-9030-1

Huang, G. B., Scheffer, L. K., and Plaza, S. M. (2018). Fully-automatic synapse prediction and validation on a large data set. *Front. Neural Circuits* 12:87. doi: 10.3389/fncir.2018.00087

Hubbard, P. M., Berg, S., Zhao, T., Olbris, D. J., Umayam, L., Maitin-Shepard, J., et al. (2020). Accelerated EM connectome reconstruction using 3D visualization and segmentation graphs. *bioRxiv [preprint]* 2020.01.17.909572. doi: 10.1101/2020.01.17.909572

Januszewski, M., Kornfeld, J., Li, P. H., Pope, A., Blakely, T., Lindsey, L., et al. (2018). High-precision automated reconstruction of neurons with flood-filling networks. *Nat. Methods* 15, 605–610. doi: 10.1038/s41592-018-0049-4

Januszewski, M., Maitin-Shepard, J., Blakely, T., Leavitt, L. J., Li, P. H., Lindsey, L., et al. (2020). A connectome and analysis of the adult Drosophila central brain. *eLife* 9:e57443. doi: 10.7554/eLife.57443

Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., et al. (2015). Saturated reconstruction of a volume of neocortex. *Cell* 162, 648–661. doi: 10.1016/j.cell.2015.06.054

Kim, J. S., Greene, M. J., Zlateski, A., Lee, K., Richardson, M., Turaga, S. C., et al. (2014). Space-time wiring specificity supports direction selectivity in the retina. *Nature* 509, 331–336. doi: 10.1038/nature13240

Kornfeld, J., and Denk, W. (2018). Progress and remaining challenges in high-throughput volume electron microscopy. *Curr. Opin. Neurobiol.* 50, 261–267. doi: 10.1016/j.conb.2018.04.030

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521:436.

Lee, K., Lu, R., Luther, K., and Seung, H. S. (2021). Learning and segmenting dense voxel embeddings for 3D neuron reconstruction. *IEEE Trans. Med. Imaging* 40, 3801–3811. doi: 10.1109/TMI.2021.3097826

Lee, K., Turner, N., Macrina, T., Wu, J., Lu, R., and Seung, H. S. (2019). Convolutional nets for reconstructing neural circuits from brain images acquired by serial section electron microscopy. *Curr. Opin. Neurobiol.* 55, 188–198. doi: 10.1016/j.conb.2019.04.001

Lee, K., Zung, J., Li, P., Jain, V., and Seung, H. S. (2017). *Superhuman Accuracy on the SNEMI3D Connectomics Challenge. ArXiv170600120 Cs*. Available online at: http://arxiv.org/abs/1706.00120 (accessed February 18, 2022).

Lee, W.-C. A., Bonin, V., Reed, M., Graham, B. J., Hood, G., Glattfelder, K., et al. (2016). Anatomy and function of an excitatory network in the visual cortex. *Nature* 532, 370–374. doi: 10.1038/nature17192

Liu, Y., and Ji, S. (2021). *CleftNet: Augmented Deep Learning for Synaptic Cleft Detection from Brain Electron Microscopy. ArXiv210104266 Cs*. Available online at: http://arxiv.org/abs/2101.04266 (accessed March 7, 2021).

Macrina, T., Lee, K., Lu, R., Turner, N. L., Wu, J., Popovych, S., et al. (2021). Petascale neural circuit reconstruction: automated methods. *bioRxiv [preprint]* 2021.08.04.455162. doi: 10.1101/2021.08.04.455162

Maitin-Shepard, J. (2021). *WebGL-Based Viewer for Volumetric Data. Goolgle*. Available online at: https://github.com/google/neuroglancer (accessed April 1, 2019).

Palacios, J., lidakanari, Zisis, E., Coste, B., MikeG, Vanherpe, L., et al. (2021). BlueBrain/NeuroM: v3.0.1. *Zenodo* doi: 10.5281/zenodo.5355891

Parekh, R., and Ascoli, G. A. (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

Peng, H., Bria, A., Zhou, Z., Iannello, G., and Long, F. (2014). Extensible visualization and analysis for multidimensional images using Vaa3D. *Nat. Protoc.* 9, 193–208. doi: 10.1038/nprot.2014.011

Peng, H., Ruan, Z., Long, F., Simpson, J. H., and Myers, E. W. (2010). V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol.* 28, 348–353. doi: 10.1038/nbt.1612

Perkel, J. M. (2019). Julia: come for the syntax, stay for the speed. *Nature* 572, 141–142. doi: 10.1038/d41586-019-02310-3

Sato, M., Bitter, I., Bender, M. A., Kaufman, A. E., and Nakajima, M. (2000). "TEASAR: tree-structure extraction algorithm for accurate and robust skeletons," in *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, (IEEE), 281–449. doi: 10.1107/S1600577516011498

Schierwagen, A. (2008). Neuronal morphology: shape characteristics and models. *Neurophysiology* 40, 310–315. doi: 10.1007/s11062-009-9054-7

Schierwagen, A., Villmann, T., Alpár, A., and Gärtner, U. (2010). "Cluster analysis of cortical pyramidal neurons using SOM," in *Proceeding of the Artificial Neural Networks in Pattern Recognition, 4th IAPR TC3 Workshop, ANNPR 2010*, (Cairo), 120–130.

Scorcioni, R., Polavaram, S., and Ascoli, G. A. (2008). L-Measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nat. Protoc.* 3, 866–876. doi: 10.1038/nprot.2008.51

Sholl, D. A. (1953). Dendritic organization in the neurons of the visual and motor cortices of the cat. *J. Anat.* 87:387.

Silversmith, W., Bae, J. A., Li, P. H., and Wilson, A. M. (2021a). Seung-lab/kimimaro: zenodo Release v1. *Zenodo* doi: 10.5281/zenodo.5539913

Silversmith, W., Collman, F., Kemnitz, N., Wu, J., Castro, M., Falk, B., et al. (2021b). Seung-lab/cloud-volume: zenodo Release v1. *Zenodo* doi: 10.5281/zenodo.5671443

Stepanyants, A., and Chklovskii, D. B. (2005). Neurogeometry and potential synaptic connectivity. *Trends Neurosci.* 28, 387–394. doi: 10.1016/j.tins.2005.05.006

Stukalov, A., and Lin, D. (2021). *Clustering.jl*. Julia Statistics. Available online at: https://github.com/JuliaStats/Clustering.jl (accessed September 30, 2021).

Turner, N. L., Lee, K., Lu, R., Wu, J., Ih, D., and Seung, H. S. (2020). "Synaptic partner assignment using attentional voxel association networks," in *Proceeding of the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, (IEEE), 1–5.

Uylings, H. B. M., and van Pelt, J. (2002). Measures for quantifying dendritic arborizations. *Netw. Comput. Neural Syst.* 13, 397–414.

Vishwanathan, A., Daie, K., Ramirez, A. D., Lichtman, J. W., Aksay, E. R. F., and Seung, H. S. (2017). Electron microscopic reconstruction of functionally identified cells in a neural integrator. *Curr. Biol.* 27, 2137.e3–2147.e3. doi: 10.1016/j.cub.2017.06.028

Vishwanathan, A., Ramirez, A. D., Wu, J., Sood, A., Yang, R., Kemnitz, N., et al. (2020). Modularity and neural coding from a brainstem synaptic wiring diagram. *bioRxiv [preprint]* 2020.10.28.359620. doi: 10.1101/2020.10.28.359620

Vishwanathan, A., Ramirez, A. D., Wu, J., Sood, A., Yang, R., Kemnitz, N., et al. (2021). Predicting modular functions and neural coding of behavior from a synaptic wiring diagram. *bioRxiv [preprint]* 2020.10.28.359620.

Wanner, A. A., Genoud, C., Masudi, T., Siksou, L., and Friedrich, R. W. (2016). Dense EM-based reconstruction of the interglomerular projectome in the zebrafish olfactory bulb. *Nat. Neurosci.* 19, 816–825. doi: 10.1038/nn.4290

Wu, J., He, Y., Yang, Z., Guo, C., Luo, Q., Zhou, W., et al. (2014). 3D BrainCV: simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution. *NeuroImage* 87, 199–208. doi: 10.1016/j.neuroimage.2013.10.036

Wu, J., Silversmith, W. M., Lee, K., and Seung, H. S. (2021). Chunkflow: hybrid cloud processing of large 3D images by convolutional nets. *Nat. Methods* 18, 328–330. doi: 10.1038/s41592-021-01088-5

Yin, W., Brittain, D., Borseth, J., Scott, M. E., Williams, D., Perkins, J., et al. (2020). A petascale automated imaging pipeline for mapping neuronal circuits with high-throughput transmission electron microscopy. *Nat. Commun.* 11:4949. doi: 10.1038/s41467-020-18659-3

Zhao, T., Olbris, D. J., Yu, Y., and Plaza, S. M. (2018). NeuTu: software for collaborative, large-scale, segmentation-based connectome reconstruction. *Front. Neural Circuits* 12:101. doi: 10.3389/fncir.2018.00101

Zhao, T., and Plaza, S. M. (2014). *Automatic Neuron Type Identification by Neurite Localization in the Drosophila Medulla. ArXiv14091892 Cs Q-Bio*. Available online at: http://arxiv.org/abs/1409.1892 (accessed November 18, 2021).

Conflict of Interest: HS has financial interests in Zetta AI LLC. This study received assistance from Google, Amazon, and Intel. These companies were not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

The remaining 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.

frontiers in Neuroinformatics

TECHNOLOGY AND CODE
published: 09 March 2022
doi: 10.3389/fninf.2022.813940

Panama: An Open-Source Educational App for Ion Channel Biophysics Simulation

Binita Rajbanshi[1†] and Anuj Guruacharya[2†]*

[1] Department of Epileptology, University Hospital Bonn, Bonn, Germany, [2] Department of Biology, University of Oklahoma, Norman, OK, United States

This article describes an open-source educational software, called Panama, developed using R, that simulates the biophysics of voltage-gated ion channels. It is made publicly available as an R package called Panama and as a web app at http://www.neuronsimulator.com. A need for such a tool was observed after surveying available software packages. Available packages are either not robust enough to simulate multiple ion channels, too complicated, usable only as desktop software, not optimized for mobile devices, not interactive, lack intuitive graphical controls, or not appropriate for educational purposes. This app simulates the physiology of voltage-gated sodium, potassium, and chlorine channels; A channel; M channel; AHP channel; calcium-activated potassium channel; transient-calcium channel; and leak-calcium channel, under current-clamp or voltage-clamp conditions. As the input values on the app are changed, the output can be instantaneously visualized on the web browser and downloaded as a data table to be further analyzed in a spreadsheet program. This app is a first-of-its-kind, mobile-friendly, and touchscreen-friendly online tool that can be used as an installable R package. It has intuitive touch-optimized controls, instantaneous graphical output, and yet is pedagogically robust for educational purposes.

Keywords: Hodgkin–Huxley simulation, web app for neuroscience, educational purposes, ion channels, biophysics

OPEN ACCESS

Edited by:
Dezhe Z. Jin,
The Pennsylvania State University (PSU), United States

Reviewed by:
Peter Ruben,
Simon Fraser University, Canada
Jeremiah Osteen,
Vertex Pharmaceuticals, United States

***Correspondence:**
Anuj Guruacharya
anuj2054@gmail.com

[†] These authors have contributed equally to this work

Received: *12 November 2021*
Accepted: *27 January 2022*
Published: *09 March 2022*

Citation:
Rajbanshi B and Guruacharya A (2022) Panama: An Open-Source Educational App for Ion Channel Biophysics Simulation. Front. Neuroinform. 16:813940. doi: 10.3389/fninf.2022.813940

INTRODUCTION

The Hodgkin–Huxley (Hodgkin et al., 1952) model is one of the fundamental neuronal models. Its mathematical form is a set of differential equations that are used before moving on to more complex models. Computational simulations using this model strengthen the concepts of action potentials and ion channels.

Existing simulation programs, such as NEURON (Hines and Carnevale, 1997) and GENESIS (Bower et al., 2003), serve as powerful tools for simulating the response of whole-cell or single-channel parameters to electrical or pharmacological stimuli. Although such software tools are free and could be used for educational or research purposes, they require substantial training and may not be suitable for casual use by students with less computational knowledge. Some effort has been directed toward making educational packages that demonstrate ion channel biophysics that is freely available. These are good tools to know about action potentials, ion channel currents, and voltages. However, each of these tools has its own disadvantages. Some of them have been highlighted below.

HHsim[1] (Touretzky et al., 2003) requires the software to be downloaded and a matching version of MATLAB installed on the desktop computer. Neurophysiology Virtual Lab[2] (Sridharan et al., 2016) requires a signup procedure and is not mobile-friendly. NeuroLab[3] (Schettino, 2014) requires a special software environment called Netlogo. Others, such as Phet,[4] are cartoon reconstructions of ion channel physiology with restricted features. Nerve[5] is not touch- or mobile-optimized whereas other programs (Molitor et al., 2006) are MATLAB packages. Any software package that is dependent on MATLAB is not ideal for wide distribution because of the overwhelming cost of MATLAB and the requirement to preinstall MATLAB. Also, any software package dependent on Java in the browser is not ideal because of the unavailability of built-in Java support in some modern web browsers.

Thus, a need was felt to make a tool that had the following characteristics: (1) mobile-friendly, (2) touchscreen-friendly, (3) pedagogically adequate for neurophysiological education, (4) completely online, (5) not reliant on MATLAB or Java software, (6) built with open-sourced code, and (7) usable by students that want an intuitive way to change ion channel parameters and download the data. To date, no electrophysiology simulation tool exists that satisfied all these criteria.

In this article, a new web app for simulating the biophysics of voltage-gated ion channels is described. It has been made publicly available at http://www.neuronsimulator.com and as a downloadable R package called Panama through GitHub. Its associated scripts are available at https://github.com/anuj2054/panama. R software is available at https://www.r-project.org/. Shiny Server is available at https://www.rstudio.com/products/shiny/. Lattice software is available at https://cran.r-project.org/web/packages/lattice/index.html. The design of the Panama software overcomes the limitations of previous simulators and satisfies all the criteria listed previously. R (R Core Team, 2014), Shiny package (Chang et al., 2017), and Lattice package (Sarkar, 2008) were used to code the software. It has multiple input controls for both voltage-clamp and current-clamp conditions. It outputs the voltage, current, and conductance values as graphs for each ion channel.

METHODS

Numerical Design of the Simulator

The 11 channels simulated in this app were voltage-gated sodium, potassium, and chloride channels; calcium-activated potassium channels (*KCa*); T-type calcium channels (*CaT*); L-type calcium channels (*CaL*), leak sodium (*NaLeak*), and leak potassium (*KLeak*) channels; A current channels; M current channels; and AHP current channels. Each channel was represented by its maximal conductance or permeability (g_n or p_n where n is the specific ion channel), its ionic current (I_n), its reversal potential (E_n), and its associated gating parameters. Total ionic current (I_{net}) was modeled as the sum of all those individual Hodgkin–Huxley style ionic currents: i_{Na}, i_K, i_{Cl}, i_{NaLeak}, i_{KLeak},i_A, i_M, i_{KCa}, i_{AHP}, i_{KCa}, i_M, i_{CaT}, and i_{CaL}. The models for these channels were modified from those used in the EOTN software (Campbell, 1996).

Voltage or current across the membrane was held constant depending on the clamping conditions. For the current-clamp case, I_{net} was held for the clamp duration at the applied current provided by the user; V_{net} was determined from Kirchhoff's current law by solving a differential equation given in Equation 1. The membrane capacitance per area represented by C in Equation 1 is input by the user and set to a default value of 0.01 nFarads.

$$\frac{dV_{net}}{dt} = \frac{1}{C}(I_{net} - i_{Na_{leak}} - i_{K_{leak}} - I_{Na} - i_K - i_{Cl} - i_{CaL} - i_{KCa} - i_{\_CaT} - i_A - i_{AHP} - i_M) \quad (1)$$

For the voltage-clamp case, V_{net} was held for the clamp duration (set to a default of 50 ms) at the applied voltage provided by the user; I_{net} was determined from Kirchhoff's current law as shown in Equation 2.

$$I_{net} = (-i_{K_{leak}} - i_{K_{leak}} - I_{Na} - i_K - i_{Cl} - i_{CaL} - i_{KCa} - i_{\_CaT} - i_A - i_{AHP} - i_M) \quad (2)$$

The default simulation time was set to 70 ms, with 10 ms being the preclamp and 10 ms being the postclamp duration. This helped to avoid data overflow issues. However, the clamp duration, preclamp duration, and postclamp duration can be changed to increase the total simulation time to 1,000 ms.

In all models, voltage was measured in *mVolt*, current in *nAmp*, time in *msec*, conductance in μ*Siemens*, and capacitance in *nFarads*. We derived the model parameters for sodium channels from data of other groups (Huguenard et al., 1988). The potassium-channel model used was of a general delayed rectifier. T-type calcium channel (i_{CaT}) was modeled using the constant field equation. L-type calcium channel (i_{CaL}) was also modeled using the constant field equation as in i_{CaT}, except that it was considered to not inactivate. i_{CaL} was based upon the data of another team (Kay and Wong, 1987) from isolated hippocampal pyramidal cells. Calcium-activated potassium channel (i_{KCa}) was modeled according to the procedure used by other groups (Yamada, 1989). We also used the same group (Yamada, 1989) to model the AHP current according to the model in bullfrog sympathetic cells. The A current was modeled to inactivate with two-time constants. The first component consisting of $m1_A$ and $h1_A$ contributes 60% to the total value of the gating variables. The second component consisting of $m2_A$ and $h2_A$ contributes 40% to the total value of the gating variables. We adapted the M current model from other groups (Adams et al., 1982).

[1]http://www.cs.cmu.edu/~dst/HHsim/

[2]http://vlab.amrita.edu/?sub=3&brch=43

[3]http://sites.lafayette.edu/schettil/neurolab/

[4]https://phet.colorado.edu/en/simulation/neuron

[5]http://nerve.bsd.uchicago.edu/

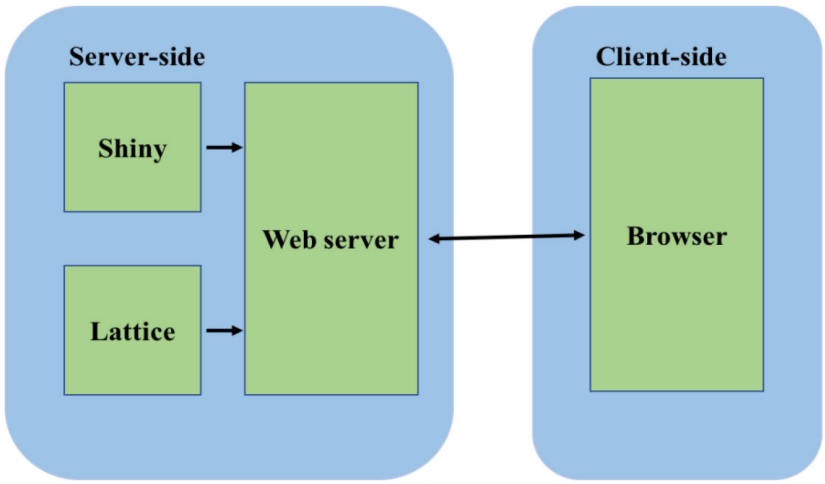


FIGURE 1 | Software architecture of the simulator. The server side consists of the Shiny package and the Lattice package. The Shiny package is used for server-side computations. The Lattice package is used for graphical display. The computations and the graphs are served to the client browser through the webserver built into the Shiny package.

Software Design of the Simulator

The web app was created using the R-programming language. After an initial survey of different languages and packages available in each language, the R language was chosen for its availability of Shiny and Lattice packages which are both excellent packages for web development and graphics development.

Euler's method was the mathematical algorithm used to solve the differential equations. The differential equations were coded into the script using only R, without using any external differential equation solver packages, such as deSolve.

The Shiny package was used to serve the webpages. The Twitter bootstrap toolkit was used as the theme for user interface controls. Sliders from the bootstrap UI toolkit were used to make the input controls touch-friendly, so that users do not have to type the values in a textbox.

The Lattice package was used to create the graphs that were embedded into the webpage. The output of the web app is a set of voltage, current, and conductance graphs for the channels. These can be visualized instantaneously while changing the input values on the app after pressing the update button, or they can be downloaded as CSV tables and analyzed in spreadsheet software. The code is open-sourced and deposited at http://www.github.com/anuj2054/panama. The front end of the software is coded in a file *ui.R*, and the backend is coded in a file server as shown in **Figure 1**.

The app is hosted on a Shiny server located at the high-performance computing center facilities at Oklahoma State University. The computations for the equations all occur on the server's side, so that there is no load on the user's computer.

RESULTS AND DISCUSSION

There are three ways to access Panama. The first and the easiest way to access it is at http://www.neuronsimulator.com/. A second way that does not require a constant internet connection to work with the software is using the command runUrl[6] on the R terminal. The command downloads the required files into an R folder and executes the software from the user's local computer. A third way to access the app is using the command shiny::runGitHub ("panama," "anuj2054") on the R terminal given that the user has preinstalled Shiny. In the second and third methods, once the required command is run and the code is automatically downloaded to the local computer, access to the internet is not required anymore.

On a desktop web browser, the input controls appear as in **Figure 2**.

However, on a mobile device, the three columns of the input controls are merged into one column for easy scrolling. On a mobile device, the use of sliders eases the process of entering values for individual parameters of the model. However, precision is not sacrificed. The user can change the input parameter values by hovering above the slider and using the keyboard to fine-tune the exact number they want to three significant decimal places. The default values on the app can be reloaded by refreshing the web browser.

The app outputs conductance, voltage, and current data as both a graphical display and as downloadable CSV tables. The ability to download the output data as CSV tables enables the user to use their own spreadsheet software, such as Excel, to further analyze the data or embed the graphs in their own documents. On a desktop web browser, the output graphs appear as shown in **Figure 3**.

Each of the lines inside the graphs is color-coded and described with the name of the channel inside its respective colored rectangle. During the current-clamp mode, the current injection can be made more noticeable in the current graph by increasing the applied current and observing the steep red

[6]https://github.com/anuj2054/panama

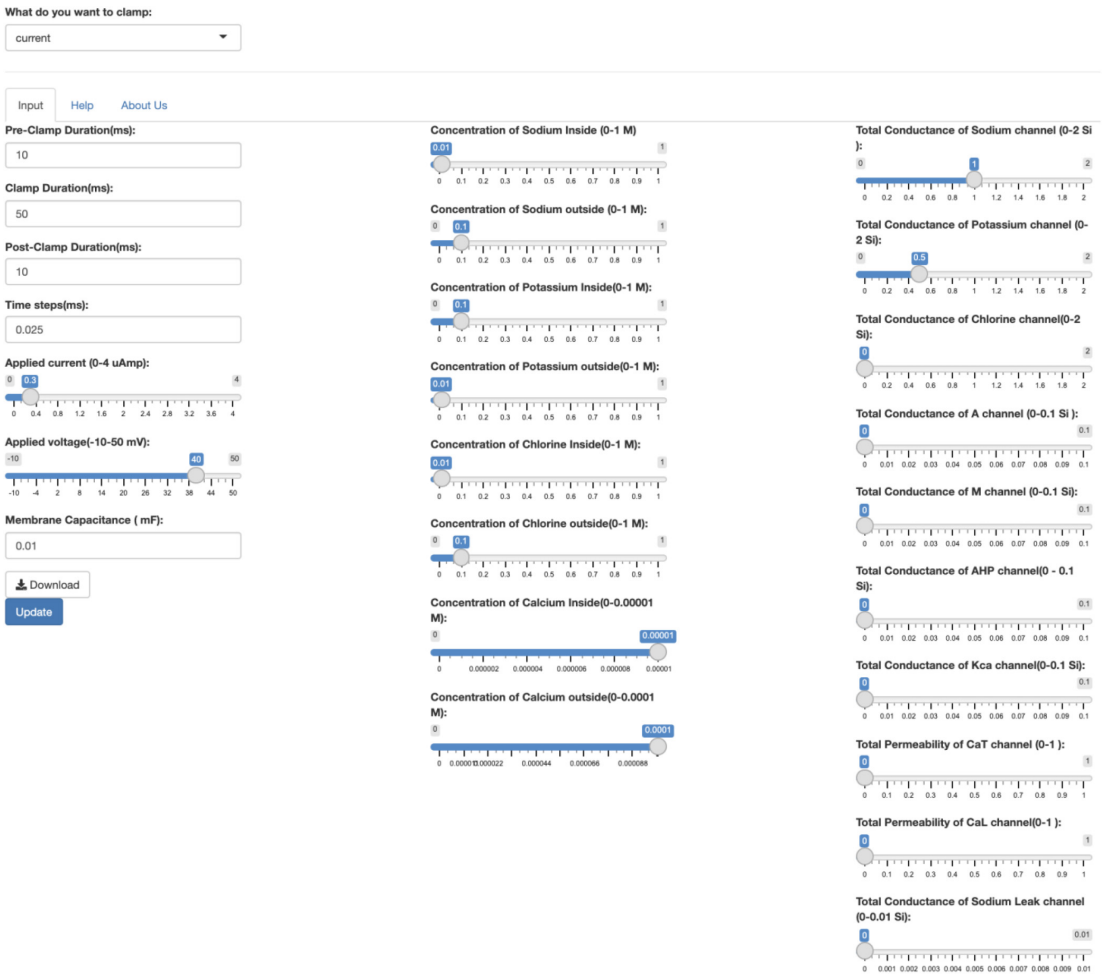


FIGURE 2 | The input controls of the app on a desktop. The user first must select the current clamp or voltage clamp conditions from the drop-down menu. After selecting the clamping conditions, various parameters can be changed using the scrollbar. Pressing the Update button will update the output graphs depending on the values input. The user must click on Update each time they change the parameters. Clicking on the Download button will download the time series voltage, current and conductance of the ion channels in a CSV format.

line that appears after the preclamp duration. The user can increase the clamp duration to see numerous action potentials that resemble neural spikes.

When using the app for educational purposes, it is encouraged to use it in conjunction with a textbook about ion channels and their electrical properties. The default ion concentrations and channel conductance can be changed to those for different types of cells, such as squid axon cells, and observe its effect on the current and voltage of the cell. The default values used in the app are for a mammalian cell at room temperature (Lodish et al., 2008). This app has been particularly helpful in pointing out the reasons for the changes observed in the action potentials when different types of channels are activated. Changing the default capacitance and conductance values demonstrates how different conditions of a cell membrane affect the electrical properties of the cell. This practice gives a hands-on approach toward learning neurophysiology that would otherwise only get from a textbook or from an expensive electrophysiology rig.

The numerical output of the simulator was tested against NEURON with similar parameters. Both programs returned equivalent results. The app was also tested under different operating systems (Windows, Android, iOS, Mac, and Linux) and under different browsers (Chrome, Firefox, and Internet Explorer). It was found to operate consistently across all platforms. This app can be used by educators, students of pharmacology, physiological science, neurobiology, and neuroscience, who are interested in simulating particular ion channels and in knowing their physiological properties so that it can be used to understand the physiological properties of voltage-gated ion channel, which acts as a triggering signal for various pathological conditions.

Future versions of the app will have phase space graphs to help users better understand membrane dynamics. It will also model synaptic currents where the chloride channels would play an important role. A better help section, tutorials, and an even cleaner user interface is also being planned.

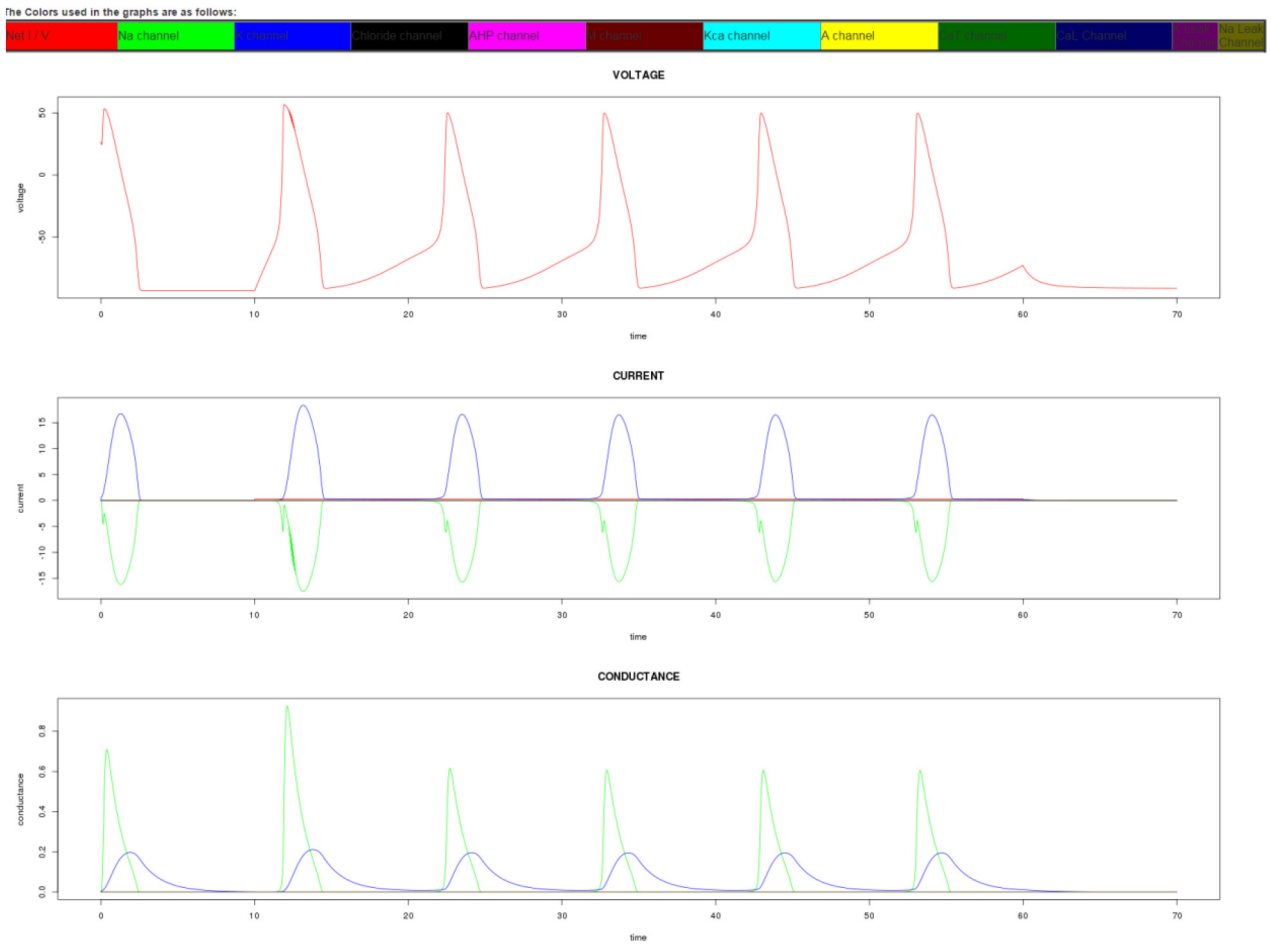


FIGURE 3 | Output graphs of the app on a desktop. The simulator gives as output the time series representation of voltage, current, and conductance. The lines in the graph are color coded to represent different ion channels. In this graph, only sodium and potassium channels have been activated. The output graphs can also be replicated by downloading the CSV file and using a spreadsheet program to draw the graphs.

Panama is a first-of-its-kind, a touch-friendly, mobile-friendly online tool that models electrophysiology of 11 different types of ion channels using Hodgkin–Huxley-style differential equations. It requires no user training for installation and no infrastructure for downloading, which makes it suitable for educational purposes. Virtual learning has been an important pedagogical tool in the physical and biological sciences. Even though the science of physics and chemistry has benefitted from having a wealth of virtual simulation tools for education, biology still lags behind in the use of such tools for education. Such portable apps can act as a virtual laboratory where there is a lack of physical resources in classrooms to purchase a high-cost electrophysiology workstation. Virtual simulation environments can improve the quality of education by providing computer-based skills in developed countries at a minimal cost. In future, such apps can provide beneficial web training by means of Massive Open Online Courses (MOOCs). Such tools should also be expanded for other areas of biological studies, such as cell biology, ecology, and population studies.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/**Supplementary Material**.

AUTHOR CONTRIBUTIONS

AG and BR conceived, designed the software, performed the analysis, and wrote the manuscript. Both authors contributed to the article and approved the submitted version.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fninf.2022.813940/full#supplementary-material

REFERENCES

Adams, P., Brown, D., and Constanti, A. (1982). M-currents and other potassium currents in bullfrog sympathetic neurones. *J. Physiol.* 330, 537–572. doi: 10.1113/jphysiol.1982.sp014357

Bower, J. M., Beeman, D., and Hucka, M. (2003). "The GENESIS simulation system," in *The Handbook of Brain Theory and Neural Networks*, ed. M. A. Arbib (Cambridge, MA: MIT Press), 475–478.

Campbell, G. (1996). Electrophysiology of the Neuron, a companion to gm shepherd's neurobiology, an interactive tutorial-by J. Huguenard and DA McCormick, Oxford University Press, 1994. &13. 95 (74 pages+ 1 computer disk, Mac and PC versions available) ISBN 0 19 509167 1. *Trends Neurosci.* 19:155.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2017). Shiny: web application framework for R. *R Pack. Vers.* 1:2017.

Hines, M. L., and Carnevale, N. T. (1997). The NEURON simulation environment. *Neural Comput.* 9, 1179–1209. doi: 10.1162/neco.1997.9.6.1179

Hodgkin, A. L., Huxley, A. F., and Katz, B. (1952). Measurement of current-voltage relations in the membrane of the giant axon of Loligo. *J. Physiol.* 116, 424–448. doi: 10.1113/jphysiol.1952.sp004716

Huguenard, J. R., Hamill, O. P., and Prince, D. A. (1988). Developmental changes in Na+ conductances in rat neocortical neurons: appearance of a slowly inactivating component. *J. Neurophysiol.* 59, 778–795. doi: 10.1152/jn.1988.59.3.778

Kay, A., and Wong, R. (1987). Calcium current activation kinetics in isolated pyramidal neurones of the Ca1 region of the mature guinea-pig hippocampus. *J. Physiol.* 392, 603–616. doi: 10.1113/jphysiol.1987.sp016799

Lodish, H., Berk, A., Kaiser, C. A., Kaiser, C., Krieger, M., Scott, M. P., et al. (2008). *Molecular Cell Biology*. Basingstoke: Macmillan.

Molitor, S. C., Tong, M., and Vora, D. (2006). MATLAB-based simulation of whole-cell and single-channel currents. *J. Undergrad. Neurosci. Educ.* 4:A74.

R Core Team (2014). *R: A Language and Environment for Statistical Computing, Version 3.1. 1*. Vienna: R Foundation for Statistical Computing.

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. Berlin: Springer Science & Business Media.

Schettino, L. F. (2014). NeuroLab: a set of graphical computer simulations to support neuroscience instruction at the high school and undergraduate level. *J. Undergrad. Neurosci. Educ.* 12:A123.

Sridharan, A., Sasidharakurup, H., Kumar, D., Nizar, N., Nair, B., Achuthan, K., et al. (2016). "Implementing a web-based simulator with explicit neuron and synapse models to aid experimental neuroscience and theoretical biophysics education," in *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*, (Delhi: Springer India).

Touretzky, D., Ladsariya, A., Albert, M., Johnson, J., and Daw, N. (2003). HHsim: an open source, real-time, graphical Hodgkin-Huxley simulator. *Soc. Neurosci. Abstr.* 29:24.13.

Yamada, W. M. (1989). "Multiple channels and calcium dynamics," in *Methods in Neuronal Modeling*, eds C. Koch and I. Segev (Cambridge, MA: The MIT Press), 97–133.

Conflict of Interest: 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.

frontiers | Frontiers in Neuroscience

BRIEF RESEARCH REPORT
published: 09 May 2022
doi: 10.3389/fnins.2022.779106

PyRAT: An Open-Source Python Library for Animal Behavior Analysis

Tulio Fernandes De Almeida, Bruno Guedes Spinelli, Ramón Hypolito Lima†, Maria Carolina Gonzalez† and Abner Cardoso Rodrigues†*

Post Graduation Program in Neuroengineering, Santos Dumont Institute, Edmond and Lily Safra International Institute of Neuroscience, Macaíba, Brazil

Here we developed an open-source Python-based library called Python rodent Analysis and Tracking (PyRAT). Our library analyzes tracking data to classify distinct behaviors, estimate traveled distance, speed and area occupancy. To classify and cluster behaviors, we used two unsupervised algorithms: hierarchical agglomerative clustering and t-distributed stochastic neighbor embedding (t-SNE). Finally, we built algorithms that associate the detected behaviors with synchronized neural data and facilitate the visualization of this association in the pixel space. PyRAT is fully available on GitHub: https://github.com/pyratlib/pyrat.

Keywords: deep learning, unsupervised learning, behavioral analysis, animal tracking, electrophysiology, neuroscience method

OPEN ACCESS

Edited by:
William T. Katz, Howard Hughes Medical Institute, United States

Reviewed by:
Brent Winslow, Design Interactive, United States
Jesse Marshall, Harvard University, United States

***Correspondence:**
Abner Cardoso Rodrigues
abner.neto@isd.org.br

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Neural Technology, a section of the journal Frontiers in Neuroscience

Received: *17 September 2021*
Accepted: *21 March 2022*
Published: *09 May 2022*

Citation:
De Almeida TF, Spinelli BG, Hypolito Lima R, Gonzalez MC and Rodrigues AC (2022) PyRAT: An Open-Source Python Library for Animal Behavior Analysis. Front. Neurosci. 16:779106. doi: 10.3389/fnins.2022.779106

1. INTRODUCTION

Deep learning (DL) and computer vision research fields are improving the performance of image, video and audio data processing (Krizhevsky et al., 2012). The use of these approaches to estimate human and animal pose is increasing rapidly. This new direction stems from several factors, including improved feature extraction, high scalability to data, availability of low-cost hardware designed for DL, and pre-trained models ready for deployment (Toshev and Szegedy, 2014; Redmon et al., 2016; Ilg et al., 2017; Levine et al., 2018; Nath et al., 2019).

Evaluation of animal behavior by human assessment is commonly subjected to inter-rater variability and requires several hours of manual video data evaluation (Spink et al., 2001). Commercial automation software for animal behavior assessment is expensive and rarely provides complex behavioral information. This software uses classical approaches of image processing to track animals' position using contrast or shape data, but they are less reliable to extract detailed information from images (Geuther et al., 2019). In contrast, DL models identify patterns in image data allowing to track the complex movement of specific body parts. Also, DL models allow 3D reconstruction of subjects using single or multiple camera setups instead of complex body markers or light sources to track positions (Nath et al., 2019; Nourizonoz et al., 2020; Dunn et al., 2021).

In the last decade, the scientific community has been incorporating DL algorithms to analyze complex behavior (Gris et al., 2017; Mathis et al., 2018; Jin et al., 2020). Usually, tracking body parts is the first step to classify and/or predict animal behavior. There are several open-source software based on DL to extract body coordinates from videos. However, they only provide the coordinate position for body parts and researchers must implement routines to infer these metrics.

Here, we present a toolbox called Python in Rodent Analysis and Tracking (PyRAT), which is a Python library capable of performing the most common analysis of animal behavior from tracking data. Our user-friendly library can integrate neural data with kinematic metrics, such as velocity, acceleration, presence in areas, and object exploration. We also implemented an unsupervised

algorithm to identify and cluster distinct animal behaviors. PyRAT is available in a public repository and can be found at: https://github.com/pyratlib/pyrat.

We believe PyRAT is a useful tool because it can be easily employed to infer some of the most common video analysis metrics through a collection of Python scripts. We developed the library to address real use cases of video analysis, frequently performed in the behavioral field. The outputs of our functions are designed to produce graphics and tables, allowing the selection of subjects and/or time window in each experiment or trial to compare groups. Other open-source libraries presents similar features, however, the behavioral community can benefit from PyRAT simpler and direct approach. We documented the library features with Jupyter notebooks in our repository to guide users to apply our code to their data.

2. MATERIALS AND METHODS

2.1. Data

To develop the PyRAT, we used datasets from the Edmond and Lily Safra International Institute of Neuroscience. Adult male Wistar rats (n = 12) were placed in an open field arena (59x59 cm with 45 cm tall walls) for 20 min per day for 3 consecutive days. Twenty-four hours later, animals were exposed to two identical objects presented in the open field arena for 5 min. We analyzed 48 videos recorded from a top-down view perspective with a Microsoft LifeCam camera at a resolution of 640 x 480 pixels at 30 frames per second (FPS). Alongside these experiments, neural data from the dorsal hippocampus were collected. All procedures were in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals and approved by a local Animal Care and Use Committee.

Furthermore, we used datasets provided by Sturman et al. (2020) and Fujisawa et al. (2008, 2015) to develop and test PyRAT functions in different scenarios. Sturman et al. (2020) used DeepLabCut to extract poses from mice in an elevated plus maze and an open field arena and provided the videos and the tracking data. Fujisawa et al. (2008) recorded single unit activity in rats performing a working memory task. The dataset is composed of extracellular recordings from the medial prefrontal cortex (64 channels) and dorsal CA1 (a subdivision of the hippocampus, 32 channels) in three rats.

2.2. Video Analysis

For body part tracking, we used DeepLabCut (DLC, version 2.2rc3) (Mathis et al., 2018; Nath et al., 2019). Specifically, we labeled 200 frames (**Figure 1A**) taken from 5 videos for each scenario (then 95% was used for training). We used a ResNet-50 neural network (Insafutdinov et al., 2016) with default parameters for 3,20,000 training iterations. We validated with 1 number of shuffles and found the test error was: 4.32 pixels, train: 2.69 pixels (image size was 640 by 480). We then used a p-cutoff of 0.9 to condition the X, Y coordinates for future analysis. This network was then used to analyze videos from similar experimental settings.

2.3. Library Design and Implementation

Our library is designed to receive as input the DLC tracking data. However, the functions work on pixel space and then can receive any tracking data after applying a few adjustments such as removing the file header, if present and renaming the columns. We developed an example using tracking data from Plexon - available on GitHUb. PyRAT was implemented using Python 3 and the following libraries: NumPy, pandas, scikit-learn, and matplotlib, and hosted in Anaconda and Python Package Index (PyPi).

2.4. Unsupervised Behavior Classification

A common task in animal behavior analysis is the identification of distinct behaviors, such as rearing, grooming, nesting, immobility, and left and right turns. To automatically classify behaviors, we used a combination of two unsupervised approaches on each video frame. We used the hierarchical agglomerative clustering algorithm to label the clusters (Lukasová, 1979) and a non-linear technique for dimensionality reduction called t-distributed stochastic neighbor embedding (t-SNE) to visualize the result (Van der Maaten and Hinton, 2008). The input of both algorithms is the distances between labeled body parts. This approach was chosen because the relative distance between body parts is invariant to the animal position in the pixel space. Combining these techniques, we created a map where the distances between the body parts of each frame are transformed into 2D space using t-SNE and the color of each point is determined by the label from hierarchical agglomerative clustering (**Figure 3A**).

To enhance cluster visualization, we optimize the t-SNE hyperparameters according to the heuristics reported in Kobak and Berens (2019). Their approach is based on three steps, (1) the use of Principal Component Analysis (PCA) in t-SNE initialization to preserve the data structure in lower dimensions; (2) set the `learning rate` as $\eta = n/12$, where n is the number of data points (frames); and (3) set the `perplexity` hyperparameter, which controls the similarity between points and governs their attraction, as $n/100$. In addition, we implemented three metrics to quantify the quality of the t-SNE output (Kobak and Berens, 2019), (1) the KNN (`k-nearest neighbors`), which quantifies the preservation of the local structure; (2) the KNC (`k-nearest class`), which quantifies the preservation of the mesoscale structure; and (3) the CPD (`Spearman correlation between pairwise distances`), which quantifies the preservation of the global structure.

Since the hyperparameters are not optimized by the learning algorithm, they must be defined *a priori* and selected by trial and error or searching approaches. However, it must be noted that these heuristics have been proven to be useful in empirical tests (Kobak and Berens, 2019).

3. RESULTS

3.1. Library Features

Python in Rodent Analysis and Tracking is a Python toolbox for the analysis of animal tracking data that is easily accessible

by new programmers, entirely developed in Python due to its popularity in the scientific community. The only prerequisite for using our toolbox is having minimal to moderate skills in Python and pandas library. We implemented the functions in a procedural approach instead of using the object-oriented features from Python as we believe that the procedural approach is more user friendly to non-programmers. Moreover, each function encapsulates an analysis, returning all inferred information and graphics. As we employed well-known Python libraries such as pandas, PyRAT can be used with other Python data science libraries such as scipy, sklearn, seaborn, matplotlib, and others.

Python in Rodent Analysis and Tracking functions receive as input a pandas DataFrame with cartesian coordinates of labeled body parts to plot the graphics (data example available on GitHub). The input format is based on the DLC output, which consists of two columns in pixel space (x and y) for each tracked body part. However, any coordinate data organized in DataFrame format can be loaded in PyRAT if it follows the structure of x and y columns for each body part.

To visualize the animal trajectory, we developed two functions. The function `Trajectory()` plots the body part coordinates across time using a matplotlib colormap (**Figure 1B**). Here, we use a scatter plot of x and y points and add a third dimension to represent time to facilitate trajectory dynamics. The other function, `Heatmap()`), generates a heatmap of the animal occupancy in the arena (**Figure 1C**). The occupancy plot is a 2D histogram that shows the body part occurrence in each spatial bin. We also evaluated the functions in a public dataset of mice performing the open field and the elevated plus maze tasks (Sturman et al., 2020).

To perform quantitative analyses, we developed the function `MotionMetrics()`, which estimates speed, acceleration, and traveled distance for each animal (**Figure 1D** and **Supplementary Material**). To estimate these metrics, we transform the data from the pixel space to the centimeters space, using a known physical reference, applying the function `pixel2centimeters()`. Also, the user can define a time interval as an input parameter to calculate the metrics (**Figure 1E**) and plot trajectory (**Figure 2A**). To test the accuracy of PyRAT functions, we used a public dataset previously analyzed with EthoVision software (Sturman et al., 2020), and we found equivalent results (data available on PyRAT's GitHub).

Experimental designs that access pathological states or drug effects can use PyRAT to extract head orientation and locomotor activity to compare treatment or conditions (Gulley et al., 2003; Aonuma et al., 2020). The function `HeadOrientation()` returns head position and orientation in each frame using two points to calculate the element-wise arc tangent between them. The head orientation must be estimated using the neck and snout; however, the same function can estimate body orientation as shown in **Figure 2B**, using the tail base and snout.

To represent the pattern of object interaction among animal groups, the `Heatmap()` function can also be used to plot concatenated data, facilitating visual comparison between days, groups, or trials (**Figure 2C**).

In addition, we developed the `FieldDetermination()` and `Interaction()` functions to evaluate the interaction of the animal with defined areas in the pixel space. For this feature, the user must first use the function `FieldDetermination()` to create circular or a rectangular area. Once the bounding areas are determined, the user must call the function `Interaction()`, which estimates animal interaction with the areas and returns a DataFrame that reports the beginning and end of each interaction in chronological order. To visualize these outputs, we developed the function `PlotInteraction()` (**Figure 2D**).

To summarize data from several subjects and facilitate visualization of behavioral metrics, we included the function `Reports()`, which combines `MotionMetrics()` and `PlotInteraction()` and creates a unified report. The input of this function is a list of the tracking data from each animal and the output is a single DataFrame (examples in **Supplementary Material**).

The function `ClassifyBehavior()` was developed to identify and classify different behaviors. We test this function in two different animal models in the open field task. In rats, 12 clusters were found automatically. The function returns a 2-dimensional color map, a histogram, and a dendrogram to better visualize the results (**Figure 3**). In addition, the histogram helps to detect mislabeled behaviors considering the number of frames in a cluster. For example, Clusters 7 and 8 presented a small number of frames, and after visual inspection, we confirmed that they were miss-classified samples (**Figure 3B**). Then, an experienced researcher must inspect the clusters to determine the type of behavior. The dendrogram shows the proximity between clusters and helps to identify the ramifications that represent a class of behavior (**Figure 3C**). In mice, 5 behavioral clusters were identified (locomotion, left/right turns, sniffing, rearing, and exploration), suggesting that PyRAT is easily generalizable to different experimental setups (data available on PyRAT's GitHub).

We developed a function to facilitate coupling the tracking data with the analysis of neural signals, in this way, we implemented the `SignalSubset()` function to extract time windows of defined events based on the interactions (function `Interaction()` output), the behavioral clusters or even from a list of timestamps (**Figure 4A**).

The function `SpatialNeuralActivity` can be used to create a map associating a neural activity to the pixel space. The input of this function is a Dataframe with the x and y of each frame together with a third column with the neural activity to be visualized. The output is a 2D NumPy array with the mean activity in each discrete space of the map. We used neural data published in Fujisawa et al. (2008) to develop an example of spike triggered activity for some units in a T-maze (**Figure 4B**). We are still developing this function to add more features, e.g., to plot the mean band of an LFP channel in the map instead of the spike data. The results and the code are available on PyRAT's GitHub.

3.2. User Guide

Python in Rodent Analysis and Tracking is a user-friendly Python toolbox to automate the analysis of animal tracking and neural data. Toolbox functions are documented, and here, we describe how to use the key features. PyRAT can be installed

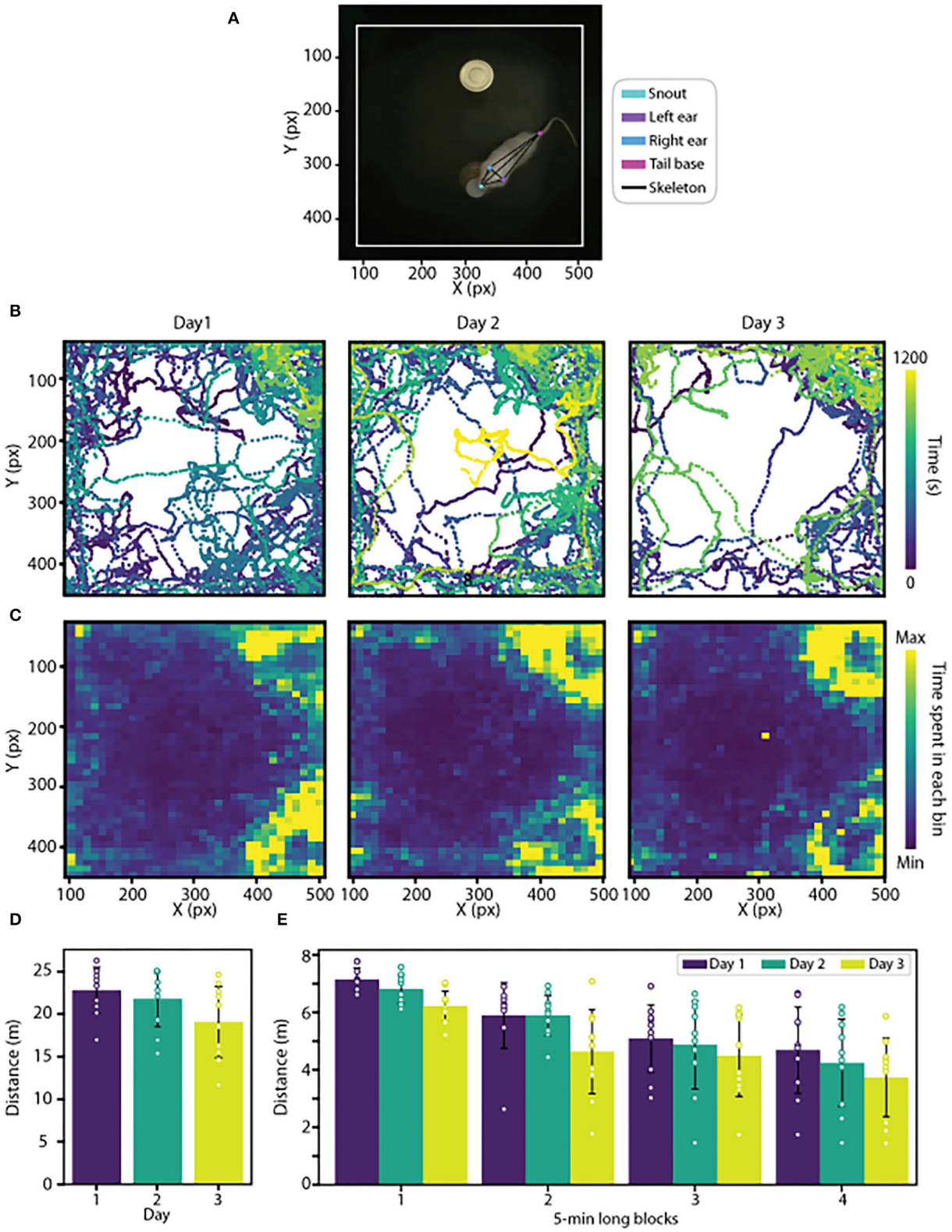


FIGURE 1 | (A) Representative image showing the marks of body parts used to train the network and the rat skeleton generated based on these marks. **(B)** Representative trajectory plots of a rat during the exploration sessions of an open field arena carried out on 3 consecutive days. Color variation indicates the moment in time at the rat's location. **(C)** Heatmaps of average trajectories during each exploration session. **(D)** Average distance traveled during each exploration session. **(E)** Average distance traveled during each exploration session is shown in blocks of 5 min per day. Data are expressed as mean ± SD.

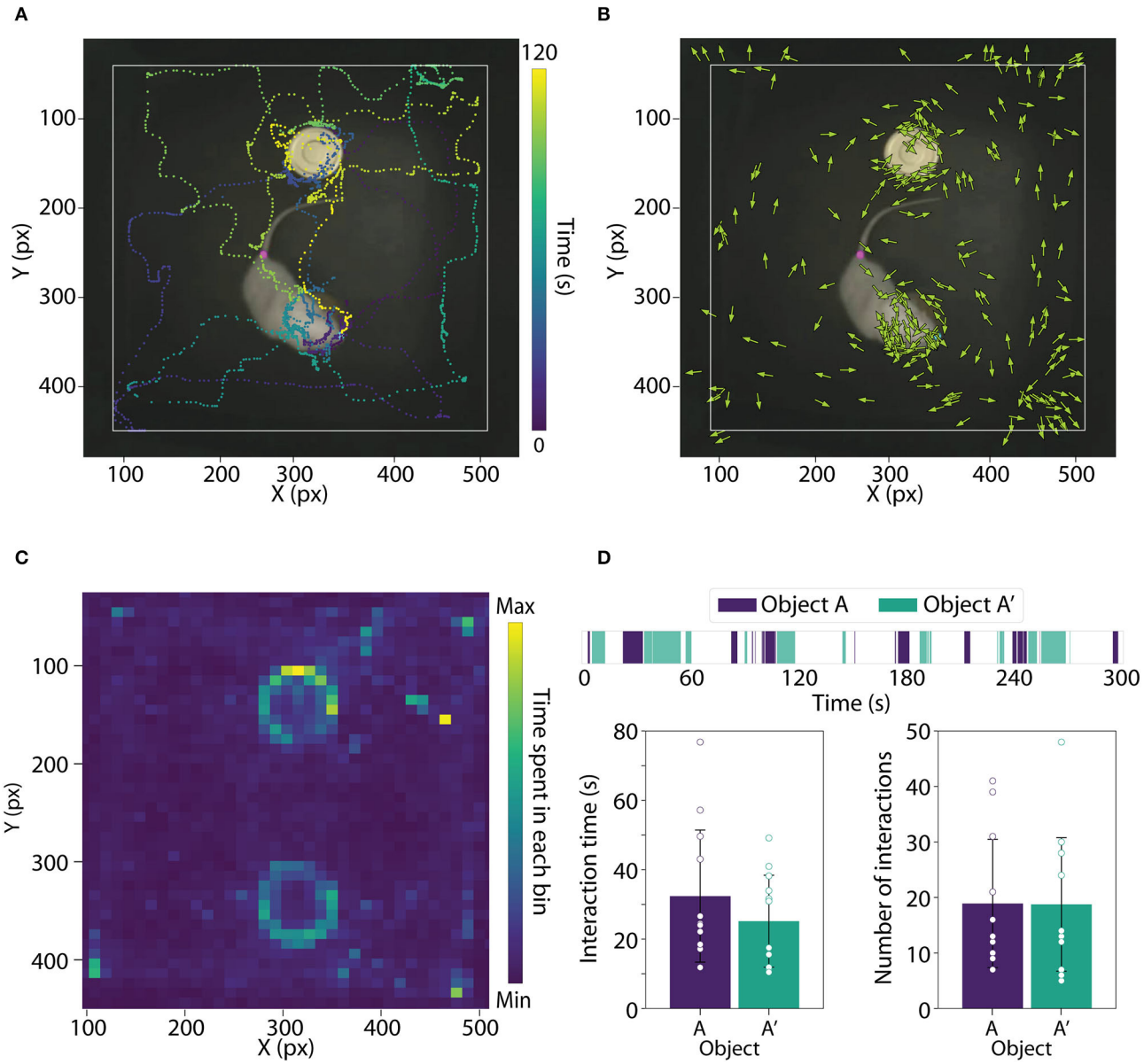


FIGURE 2 | (A) Image showing the trajectory of one rat for 120 s based on the snout coordinates. **(B)** Image showing rat body orientation during the entire object exploration session. **(C)** Average heatmap during the entire object exploration session. **(D)** Top: Object interaction across the entire object exploration session; Bottom left: Bar plot showing interaction time with objects A and A'; Bottom right: Bar plot showing the number of interactions with object A and A'. Data are expressed as mean ± SD.

using `pip install pyratlib`. Then, it is necessary to import the following libraries:

```
import pyratlib as rat
import pandas as pd
```

Subsequently, the user must read tracking data as a DataFrame, e.g., using the `read_csv()` function from pandas. This Dataframe will be used as input on the majority of PyRAT functions. Here, we show how to plot the trajectories and the heatmap:

```
data = pd.read_csv('your_data_path.csv')

rat.Trajectory(data, bodyPart = 'tail', bodyPartBox = 'tail')
rat.Heatmap(data, bodyPart = 'tail', bins = 10, vmax = 50)
```

To plot the trajectory, the user must define a body part in the function `Trajectory` using the `bodyPart` parameter which is the column name of the chosen body part. The function `Heatmap()` uses the `bodyPart` and the parameters `bins` and `vmax`, which determine the resolution and color scale of the plot.

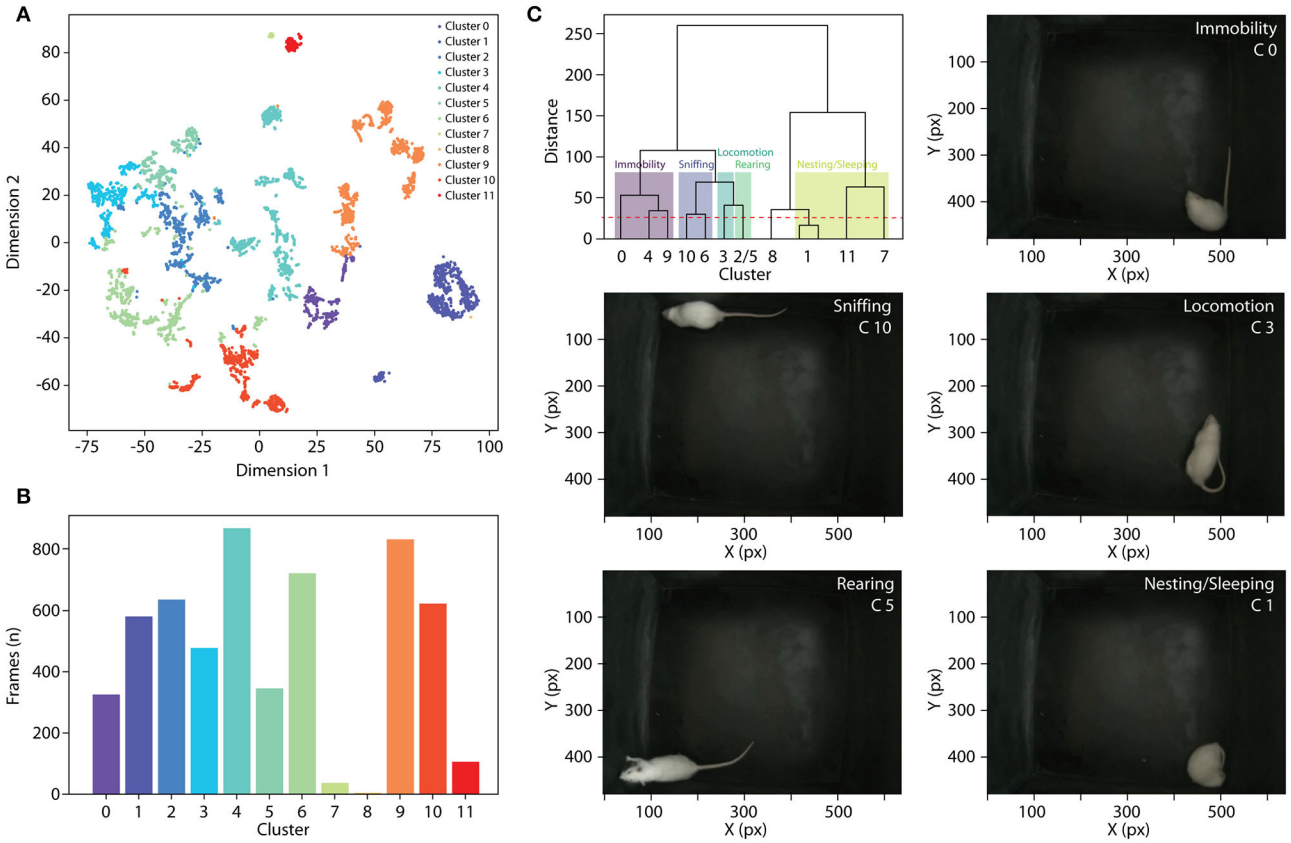


FIGURE 3 | (A) Bidimensional projection representing each cluster found by the unsupervised algorithm of behavior classification. **(B)** Histogram showing the number of frames in each cluster. **(C)** Top left: Dendrogram presenting the proximity of the clusters. Clusters with similar behaviors were grouped after visual inspection. We identified five behavioral clusters: immobility, sniffing, locomotion, rearing, and nesting/sleeping. The other images are representative frames showing some of the behavioral clusters identified.

Another PyRAT feature is the quantification of the interaction between a body part and an area. This interaction can be calculated with the function `Interaction()` and defining a bounding area by passing the size and coordinates of the vertices. The function `FieldDetermination()` allows the visualization of areas in the pixel space, according to the tracking data. Also, we developed the function `PlotInteraction()` to plot the beginning, end, and duration of interactions with each bounding area across time:

```
obj_dict = {'Obj_1': [1,0,0,0,430, 35,90,75],
            'Obj_2': [1,0,0,0,430,380,90,75]}

objects = rat.FieldDetermination(posit = obj_dict)
interactions = rat.Interaction(data,'snout',objects)
rat.PlotInteraction(interactions)
```

In the example above, two areas representing objects in distinct positions were passed as input, and the output is a DataFrame with the timestamps of each object interaction. The function `PlotInteraction()` plots object interactions across time (**Figure 2D**).

The function `ClassifyBehavior()` is a behavioral classifier and receives as parameters the tracking DataFrame, the video directory, the selected body parts, and the distance:

```
rat.ClassifyBehavior(df,
                     video = 'path',
                     bp_list = ['snout', 'ear_R', 'ear_L', 'tail'],
                     distance = 28)
```

The distance metric passed in this function is Ward's distance and defines the threshold above which the clusters will not be merged.

To facilitate the analysis of neural signals recorded during behavioral tasks, we developed the function `Interaction()` to extract timestamps of events of interest and the function `SignalSubset()` to extract epochs of the neural signal. An example of neural data input is available in **Supplementary Material**. We used files from Plexon and Blackrock Neurotech, but data from other acquisition systems can be used.

```
subsets = rat.SignalSubset(signal, freq = 1000,
                           fields = interactions)
```

`SignalSubset()` returns the extracted data organized in a dictionary with the number of the epoch as the

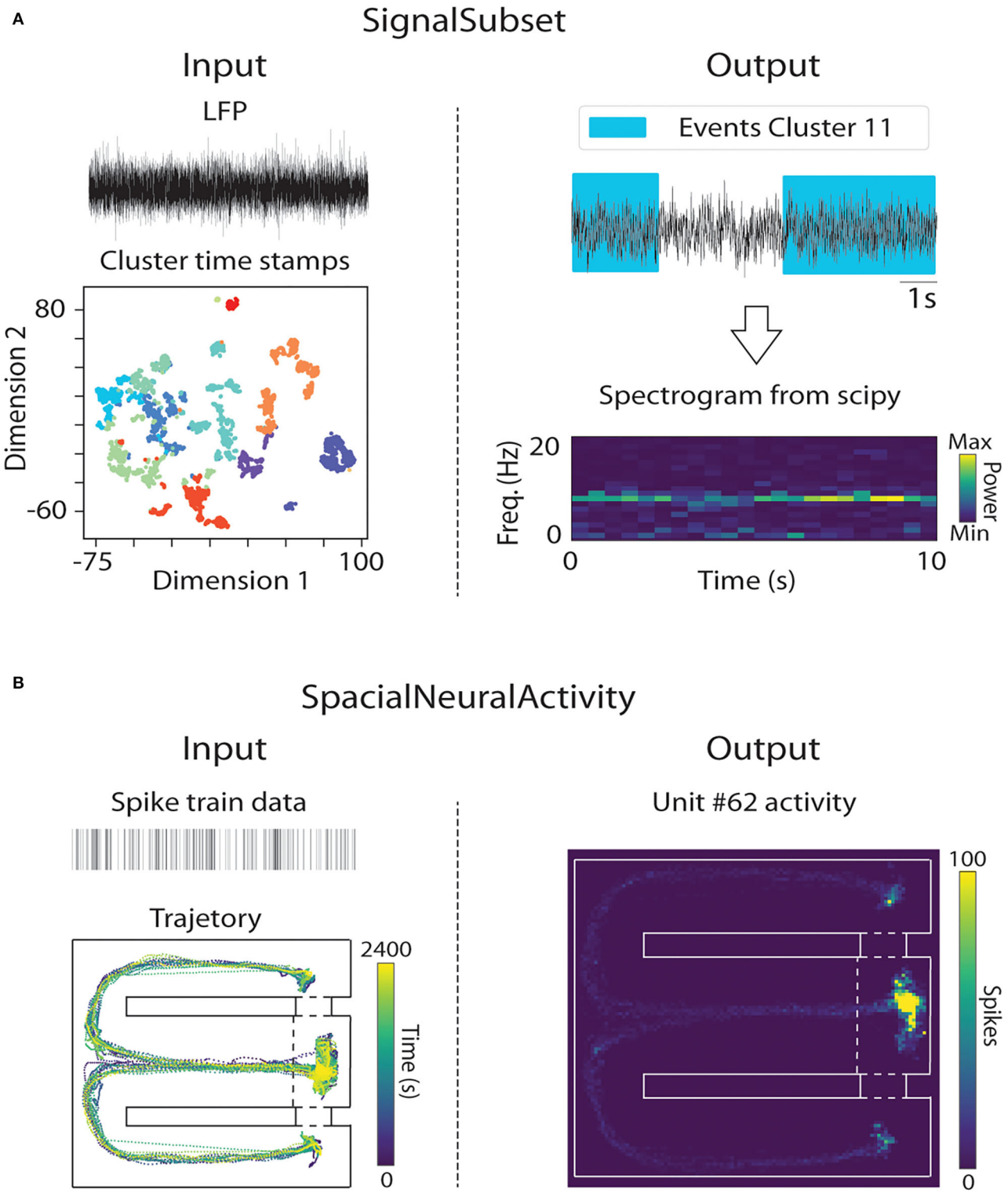


FIGURE 4 | (A) Overview of SignalSubset function. Left column: The SignalSubset function receives as input neural data (e.g., raw LFP) and the clustermap produced by ClassifyBehavior. Right column: SignalSubset function returns a list of extracted neural data corresponding to time windows of a determined behavior (e.g., Cluster 11). We also show a representative spectrogram of extracted data. **(B)** Overview of SpatialNeuralActivity function. Left column: The SpatialNeuralActivity function receives as input the neural data (e.g., single unit spike rasterplot) to be shown in pixel space, and the tracking as spatial data. Right column: The SpatialNeuralActivity function returns the quantification of neural activity (spike firing) in each part of the pixel space.

key. In addition, it can extract the time of a selected behavioral cluster. For this, it is necessary to use the cluster output from `ClassifyBehavior()` as input to the `IntervalBehaviors()` function, which will return a dictionary with the time windows when each behavior was manifested (documented in GitHub). This function facilitates data processing and allows saving the dictionary, speeding up data loading.

The function `Reports()`, which summarizes data from several animals, receives as input the lists with DataFrames and the file names, as well as the body part of interest to extract the metrics and, if necessary, an area to calculate interactions:

```
list_df = [df01,df02,df03,df04,df05,df06,
           df07,df08,df09,df10,df11,df12]
names = ['RAT01','RAT02','RAT03','RAT04','RAT05','RAT06',
         'RAT07','RAT08','RAT09','RAT10','RAT11','RAT12']
report = rat.Reports(df_list = list_df,list_name = names,
                     bodypart = 'snout',fields = objects)
```

4. DISCUSSION

We presented the PyRAT, a library for animal tracking data analysis developed to be accessible to less experienced programmers. We implemented functions to infer common animal behavioral metrics used in the literature, such as object interaction (duration and number of interactions), traveled distance, speed, and time spent in different areas (Lima et al., 2009; Gonzalez et al., 2019; Rossato et al., 2019; Moura et al., 2020). Also, we implemented functions to infer animal behavior from tracked body parts in each frame using unsupervised approaches. If video recordings are synchronized with neural data, PyRAT can be used to extract epochs based on specific behaviors or metrics. Finally, our results indicate that PyRAT analyzes tracking data from different animal models if videos were acquired from a top-down perspective.

There is similar software that can analyze tracking data as PyRAT, such as Traja, DLCAnalyzer, SimBA, and B-SOiD. Traja is a Python library that can analyze tracking data from coordinate data from any setup but does not infer behavioral metrics. DLCAnalyzer is a collection of R scripts that processes DLC files and quantifies motion metrics and behavior using supervised algorithms (Sturman et al., 2020). Simple Behavioral Analysis (SimBA) is software with an easy-to-use interface that analyzes video or tracking data and applies a pre-trained supervised classifier to cluster behaviors (Nilsson et al., 2020). However, the SimBA interface only works in Windows, limiting its usability on other platforms. B-SOiD is an open-source package that identifies behavior by combining supervised and unsupervised algorithms (Hsu and Yttri, 2021) and works in mice, rats, and humans. B-SOiD analyzes videos acquired from different perspectives, showing the best results from bottom-up recordings. For further discussion and comparison between these tools refer to Panadeiro et al. (2021); von Ziegler et al. (2021). In contrast with other tools, PyRAT can be used in any operational system, does not need pre-trained classifiers, works without a graphic interface, and provides interactive documentation using Jupyter notebooks.

Python in Rodent Analysis and Tracking is easier to use than other alternatives as it is a collection of functions, and the user just needs to input the tracking data to get the results and graphics following the step-by-step tutorial included in the documentation. In addition, PyRAT has a low learning curve, as its implementation is based on procedural programming. We designed the library to display metrics and graphics for all recorded sessions with a few lines of code. It does not have software requirements besides Python and widely used libraries, such as sklearn, pandas, and matplotlib. In summary, we present an open-source Python library to process tracking data, extract behavior and associate this information with neural data in a user-friendly approach.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are publicly available. This data can be found here: GitHub: https://github.com/pyratlib/pyrat; Zenodo: https://zenodo.org/record/5883277.

ETHICS STATEMENT

The animal study was reviewed and approved by Animal Research Ethics Committee of Santos Dumont Institute.

AUTHOR CONTRIBUTIONS

TD, BS, and AR designed, wrote, tested the library, and performed the analysis of the examples. RH and MG evaluated the algorithms. TD documented the library. TD, RH, MG, and AR wrote the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

"This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível" "Superior – Brasil (CAPES) – Finance Code 001.", Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Ministério da Educação (MEC), and Instituto Santos Dumont (ISD).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2022.779106/full#supplementary-material

REFERENCES

Aonuma, H., Mezheritskiy, M., Boldyshev, B., Totani, Y., Vorontsov, D., Zakharov, I., et al. (2020). The role of serotonin in the influence of intense locomotion on the behavior under uncertainty in the mollusk lymnaea stagnalis. *Front. Physiol.* 11, 221. doi: 10.3389/fphys.2020.00221

Dunn, T. W., Marshall, J. D., Severson, K. S., Aldarondo, D. E., Hildebrand, D. G., Chettih, S. N., et al. (2021). Geometric deep learning enables 3d kinematic profiling across species and environments. *Nat. Methods* 18, 564–573. doi: 10.1038/s41592-021-01106-6

Fujisawa, S., Amarasingham, A., Harrison, M. T., and Buzsáki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nat. Neurosci.* 11, 823–833. doi: 10.1038/nn.2134

Fujisawa, S., Amarasingham, A., Harrison, M. T., and Buzsáki, G. (2015). Simultaneous electrophysiological recordings of ensembles of isolated neurons in rat medial prefrontal cortex and intermediate ca1 area of the hippocampus during a working memory task. *Dataset* 1, 1–6. doi: 10.6080/K01V5BWK

Geuther, B. Q., Deats, S. P., Fox, K. J., Murray, S. A., Braun, R. E., White, J. K., et al. (2019). Robust mouse tracking in complex environments using neural networks. *Commun. Biol.* 2, 1–11. doi: 10.1038/s42003-019-0362-1

Gonzalez, M. C., Rossato, J. I., Radiske, A., Reis, M. P., and Cammarota, M. (2019). Recognition memory reconsolidation requires hippocampal zif268. *Sci. Rep.* 9, 1–11. doi: 10.1038/s41598-019-53005-8

Gris, K. V., Coutu, J.-P., and Gris, D. (2017). Supervised and unsupervised learning technology in the study of rodent behavior. *Front. Behav. Neurosci.* 11, 141. doi: 10.3389/fnbeh.2017.00141

Gulley, J. M., Hoover, B. R., Larson, G. A., and Zahniser, N. R. (2003). Individual differences in cocaine-induced locomotor activity in rats: behavioral characteristics, cocaine pharmacokinetics, and the dopamine transporter. *Neuropsychopharmacology* 28, 2089–2101. doi: 10.1038/sj.npp.1300279

Hsu, A. I., and Yttri, E. A. (2021). B-soid, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nat. Commun.* 12, 1–13. doi: 10.1038/s41467-021-25420-x

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 2462–2470.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). "Deepercut: a deeper, stronger, and faster multi-person pose estimation model," in *European Conference on Computer Vision* (Amsterdam: Springer), 34–50.

Jin, T., Duan, F., Yang, Z., Yin, S., Chen, X., Liu, Y., et al. (2020). Markerless rat behavior quantification with cascade neural network. *Front. Neurorobot.* 14, 570313. doi: 10.3389/fnbot.2020.570313

Kobak, D., and Berens, P. (2019). The art of using t-sne for single-cell transcriptomics. *Nat. Commun.* 10, 1–14. doi: 10.1038/s41467-019-13056-x

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25, 1097–1105.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Rob. Res.* 37, 421–436. doi: 10.1177/0278364917710318

Lima, R. H., Rossato, J. I., Furini, C. R., Bevilaqua, L. R., Izquierdo, I., and Cammarota, M. (2009). Infusion of protein synthesis inhibitors in the entorhinal cortex blocks consolidation but not reconsolidation of object recognition memory. *Neurobiol. Learn. Mem.* 91, 466–472. doi: 10.1016/j.nlm.2008.12.009

Lukasová, A. (1979). Hierarchical agglomerative clustering procedure. *Pattern Recognit.* 11, 365–381. doi: 10.1016/0031-3203(79)90049-9

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., et al. (2018). Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* 21, 1281–1289. doi: 10.1038/s41593-018-0209-y

Moura, C. A., Oliveira, M. C., Costa, L. F., Tiago, P. R., Holanda, V. A., Lima, R. H., et al. (2020). Prenatal restraint stress impairs recognition memory in adult male and female offspring. *Acta Neuropsychiatr.* 32, 122–127. doi: 10.1017/neu.2020.3

Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., and Mathis, M. W. (2019). Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nat. Protoc.* 14, 2152–2176. doi: 10.1038/s41596-019-0176-0

Nilsson, S. R., Goodwin, N. L., Choong, J. J., Hwang, S., Wright, H. R., Norville, Z., et al. (2020). Simple behavioral analysis (simba): an open source toolkit for computer classification of complex social behaviors in experimental animals. *BioRxiv*. doi: 10.1101/2020.04.19.049452

Nourizonoz, A., Zimmermann, R., Ho, C. L. A., Pellat, S., Ormen, Y., Prévost-Solié, C., et al. (2020). Etholoop: automated closed-loop neuroethology in naturalistic environments. *Nat. Methods* 17, 1052–1059. doi: 10.1038/s41592-020-0961-2

Panadeiro, V., Rodriguez, A., Henry, J., Wlodkowic, D., and Andersson, M. (2021). A review of 28 free animal-tracking software applications: current features and limitations. *Lab. Anim.* 50, 246–254. doi: 10.1038/s41684-021-00811-1

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition* (Las Vegas, NV: IEEE), 779–788.

Rossato, J. I., Gonzalez, M. C., Radiske, A., Apolinário, G., Conde-Ocazionez, S., Bevilaqua, L. R., et al. (2019). Pkmζ inhibition disrupts reconsolidation and erases object recognition memory. *J. Neurosci.* 39, 1828–1841. doi: 10.1523/JNEUROSCI.2270-18.2018

Spink, A., Tegelenbosch, R., Buma, M., and Noldus, L. (2001). The ethovision video tracking systemâĂŤa tool for behavioral phenotyping of transgenic mice. *Physiol. Behav.* 73, 731–744. doi: 10.1016/S0031-9384(01)00530-3

Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., et al. (2020). Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology* 45, 1942–1952. doi: 10.1038/s41386-020-0776-y

Toshev, A., and Szegedy, C. (2014). "Deeppose: human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 1653–1660.

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579–2605.

von Ziegler, L., Sturman, O., and Bohacek, J. (2021). Big behavior: challenges and opportunities in a new era of deep behavior profiling. *Neuropsychopharmacology* 46, 33–44. doi: 10.1038/s41386-020-0751-7

Conflict of Interest: 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.

frontiers | Frontiers in Neuroinformatics

PERSPECTIVE
published: 16 May 2022
doi: 10.3389/fninf.2022.828458

Connectomics Annotation Metadata Standardization for Increased Accessibility and Queryability

Morgan Sanchez[1†], *Dymon Moore*[1†], *Erik C. Johnson*[1], *Brock Wester*[1], *Jeff W. Lichtman*[2] *and William Gray-Roncal*[1*]

[1] *Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD, United States,* [2] *Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, United States*

Neuroscientists can leverage technological advances to image neural tissue across a range of different scales, potentially forming the basis for the next generation of brain atlases and circuit reconstructions at submicron resolution, using Electron Microscopy and X-ray Microtomography modalities. However, there is variability in data collection, annotation, and storage approaches, which limits effective comparative and secondary analysis. There has been great progress in standardizing interfaces for large-scale spatial image data, but more work is needed to standardize annotations, especially metadata associated with neuroanatomical entities. Standardization will enable validation, sharing, and replication, greatly amplifying investment throughout the connectomics community. We share key design considerations and a usecase developed for metadata for a recent large-scale dataset.

Keywords: connectome, annotation, software, standard, queries, reproducibility

OPEN ACCESS

Edited by:
Ting Zhao,
Janelia Research Campus, United States

Reviewed by:
Pat Gunn,
Flatiron Institute, United States
Kevin Boergens,
Paradromics, Inc., United States

***Correspondence:**
William Gray-Roncal
William.Gray.Roncal@jhuapl.edu

[†] *These authors have contributed equally to this work*

Received: *03 December 2021*
Accepted: *31 January 2022*
Published: *16 May 2022*

Citation:
Sanchez M, Moore D, Johnson EC, Wester B, Lichtman JW and Gray-Roncal W (2022) Connectomics Annotation Metadata Standardization for Increased Accessibility and Queryability. Front. Neuroinform. 16:828458. doi: 10.3389/fninf.2022.828458

1. INTRODUCTION

In an effort to better understand structural organization and anatomy of nervous systems at nanoscale spatial resolution, increasingly large, even petascale, connectomics datasets have been collected using Electron Microscopy (EM) and X-Ray Microtomography (XRM) (Kasthuri et al., 2015; Schneider-Mizell et al., 2020; Xu et al., 2020; Consortium et al., 2021; Shapson-Coe et al., 2021; Witvliet et al., 2021). Currently, researchers and automated algorithms can label cells, subcellular components, and connections between cells to generate brain networks. Formats of such annotations, however, can vary greatly between datasets and institutions. As such, the computational expertise required to explore large, unfamiliar datasets and understand heterogeneous raw annotations remains a serious barrier to their widespread reuse, such as for downstream analysis of previously-collected and potentially unfamiliar data. Consequently, there is demand for simple community-adopted standards for storing key information about neuroanatomical entities represented in EM, XRM, and correlated light microscopy (LM) datasets as well as software tools built upon these standards to allow any researcher to quickly and easily extract information on annotated bodies without grappling with raw annotation downloads and lab-specific post-processing pipelines. This work focuses on filling a key need for the community by addressing a central aspect of annotation variability. It calls for standardized storage of metadata associated with key neuroanatomical entities, such as neurons, synapses, and organelles to supplement raw annotations. It also suggests an approach to metadata standardization through the use of community-adopted definitions, and demonstrates an example of how such standards can facilitate the development of simple data exploration interfaces.

Raw annotations may have any or all of the following formats: segmentations, anatomical skeletons and meshes, synaptic connectivity networks, and information in the form of tables or network attributes. From dataset to dataset, each of these primary data formats and associated documentation can vary in terms of structure, meaning, and transparency, making it difficult to use them to accurately extract relevant information about commonly-studied entities and test simple hypotheses that may not rely on spatial representations of the data. This motivates the use of standardized formats for storing annotated objects along with key attributes, separate from a lab's chosen raw annotation format, which may store such information indirectly. (i.e., The number of synapses on a neurite can be extracted through segmentation post-processing).

In this period of growth in EM, XRM, and correlated LM imaging, and their increased adoption and utilization in neuroscience, it would be advantageous to implement standards that ensure interoperability and sustainability, beyond just availability of these datasets through public release. This will promote rapid analysis, true openness, sharing between laboratories, and reproducible results in connectomics research. Existing annotation formats serve their purposes within labs, but extraction of neuroanatomical entities and properties in a standardized format can facilitate cross-institutional collaboration and exploration that existing formats do not always permit. Further, such standardization will enable the development of existing, as well as additional shared computational tools with user-friendly interfaces for querying these unique data for scientific discovery regardless of a user's computational expertise.

Because these datasets are large and complex, it is especially important to promote data exploration and discovery. Visualization and querying tools exist already, but are often lab- or dataset-specific (Clements et al., 2020). Furthermore, developers of new software must choose a data representation to support, which limits each new tool's broad applicability. One benefit of annotation standards is the potential for mitigation of this challenge through design and modification of software tools to build upon annotation standards. Visualization and querying software such as Neuroglancer (Maitin-Shepard, 2020), Neuromorpho (Ascoli et al., 2007), DotMotif (Matelsky et al., 2021), Webknossos (Boergens et al., 2017), NeuPrint (Clements et al., 2020), and others (Yatsenko et al., 2015) can be modified to support community-developed annotation standards and even integrated into a standards-supported, centralized discovery portal geared toward users without extensive computational backgrounds. Such a centralized connectomics discovery platform that allows exploration of datasets across imaging modalities, organisms, and institutions, is an exciting prospect, and is most feasible once metadata standardization is adopted.

This work will discuss the need for annotation metadata standards, propose a framework for such standardization, and demonstrate an application of such standards. To demonstrate potential impacts of standards on analysis software, we provide a case study in which we build tools to store and query a large emerging human connectome dataset, H01 (Shapson-Coe et al., 2021).

2. ANNOTATION STANDARDS

An acknowledged challenge in the field of connectomics is mitigating the impact of highly varied annotation representation on software and institution-level interoperability (Plaza et al., 2014). As the field grows and data volumes increase, the necessity for sharing data through remote and programmatic interfaces increases, and, in turn, the need for community-developed algorithms and software to extract and process that data also grows. Answering this challenge requires creating and popularizing annotation representation standards which enable parsing and understanding the scenes present in these nanoscale neuroimaging volumes, without alienating researchers with existing analysis pipelines.

Because the fields of EM and XRM data are still emerging, defining standards for these communities is timely. In order to enable community-oriented connectomics frameworks and collaboration, new annotation standards and software tools built to support those standards must strike a balance between organization and flexibility which is why we focus on standardized, expandable neuroanatomical entity definitions to store metadata as opposed to restricting raw annotation formats.

2.1. Support Common Raw Annotation Formats

The call for metadata standardization does not necessitate abandonment of existing raw annotation representation formats. Abandoning these formats could lead to obsolescence of existing, useful annotation, and analysis software (Ascoli et al., 2007; Saalfeld et al., 2009; Boergens et al., 2017; Berger et al., 2018) and ultimately alienation of institutions with incompatible formats, and is not the focus of this article. Instead, we hope to provide a blueprint for a new export format and urge institutions to build import/export tools. New standards, therefore, can continue to support a variety of common data representations.

Though not the focus of this article, the authors recognize that raw annotation formats could benefit from improvements as well, specifically in terms of documentation. Further work could better connect metadata to raw annotations and convey how neuroanatomical entities are represented in raw annotation data.

2.2. Facilitate Connections Between Datasets

Additionally, community-adopted annotation standards can enable linkage between data modalities and datasets. This facilitates comparison, meta-analysis, and registration with other datasets and imaging modalities. Links to different data modalities such as those between structural and functional LM data for the same subject, can encourage exciting research relating structure to function at the synaptic level (Consortium et al., 2021), and links between datasets can facilitate analysis across brain regions, individuals, and species, paving the way for understanding what is conserved and what differs across datasets and enabling large-scale discovery.

2.3. Metadata Standardization

Storage of metadata associated with neuroanatomical entities needs to be standardized to promote reproducibility, extensibility, and queryability of connectomics metadata. As such, new metadata standards must be built around the core knowledge products extracted from neurons, synapses, and their relationships (e.g., connectivity). Further, because user needs for data processing are diverse, standards must be conducive to common nanoscale connectomics research questions, such as those pertaining to location, topology, morphology, and cell types (LaGrow et al., 2018) as well as those surrounding connectivity at a local or circuit level (Matelsky et al., 2021) and even at higher-levels pertaining to brain regions and white matter tracts (Sporns et al., 2004; Bassett and Bullmore, 2006).

To satisfy diverse user needs as well as the need for standardization, we propose that the community agree upon definitions and minimum required attributes for key entities extracted from connectomics data, which are relevant to a variety of research areas. In **Table 1**, we define several areas and examples for annotation metadata standardization. For each neuroanatomical entity in the dataset (e.g., neuron, synapse, neurite, etc.) data owners should provide the URI, data representation, type, and, when applicable, links to other entities. Additionally, entities can be either user-defined or community-defined. Though data owners have the option to define new entities (e.g., user-defined), there are several entities which should have community-adopted definitions and properties. This combination of entities appropriately balances structure and flexibility in a way that allows software built upon standards to extract information uniformly across datasets while also allowing researchers to store additional non-standardized entities and metadata as desired. It also allows researchers to choose the level of granularity at which to store a dataset. Though community definitions will exist for multiple levels (e.g., from vesicles and mitochondria to neurons and layers), not every dataset needs to include all of these entities. Larger datasets, for example, may only include higher-level entities, while smaller datasets might contain lower-level entities. The only stipulation is that if a dataset chooses to include a particular entity, that entity's minimum properties must be satisfied.

Our approach to annotation representation and metadata follows a neuroscience schema, previously used internally, called Reusable Annotation Markup for Open Neuroscience (RAMON) (Gray Roncal et al., 2015). RAMON defines a minimum set of annotation types and associated metadata that capture important biological information as well as relationships between annotations that are critical for connectome generation and neuroscientific exploration.

In particular, the H01 synapse annotation type includes metadata such as synapse id, type, and associated neurons. Currently, RAMON defines metadata standards for biological entities which are used commonly across connectomics datasets, such as neurons, synapses, and organelles, although this can be extended to additional entity types as needed.

3. ANNOTATION QUERIES

As mentioned previously, one benefit of metadata standardization is that it enables the development of tools to query data, regardless of its origin. Through queries, researchers can characterize networks, extract patterns, and relate these patterns to function. Currently, however, asking even basic, fundamental questions (e.g., how many, how much) about a new dataset can be challenging from both a standardization and computational complexity perspective. Though previous work has presented information extraction tools for specific datasets and institutions (Clements et al., 2020), metadata standardization has the potential to expand the use of existing tools cross-institutionally, foster the development of new ones, and facilitate integration of numerous tools into a single location. The community would benefit from a shared discovery portal built upon community archives and standards (Ascoli et al., 2007; Sunkin et al., 2012; Vogelstein et al., 2018; Rübel et al., 2019), which provides broad accessibility to EM and XRM data and annotations through query submission tools to enable deeper understanding of these data. In this work, we demonstrate this particular benefit of metadata standardization through the development of a simple querying tool built upon RAMON.

Ideally, researchers should be able to easily query counts, distributions, properties, and connectivity of neuroanatomical entities as well as image and graph metrics for any connectomics dataset, regardless of source institution. Queries such as number of synapses in a given region, or the distribution of synapses on a particular neuron type could help answer a variety of research questions, but the broad community interested in brain atlases and neuroanatomy has traditionally had little access to and experience with large-scale EM and XRM datasets. Tools for executing standardized queries could, therefore, enable a new wave of discoveries.

4. CASE STUDY: H01 HUMAN DATA

Here, we present a case study, in which we store annotations from a petascale human cortex dataset, the H01 dataset (Shapson-Coe et al., 2021) and build tools to that allow users to access and query that data through a web application. The H01 dataset consists of a cubic millimeter volume with annotations for 50,000 cells, hundreds of millions of neurites, and 150 million synapses, taken from a human surgical sample from the temporal lobe of the cerebral cortex. This dataset was chosen because of its size, breadth of annotation types, and significance as the first large, nanoscale human connectomics dataset. To demonstrate the robustness and generalizability of our approach, we also include a second dataset (Kasthuri et al., 2015) in our database and query engine.

4.1. Software Architecture

As a demonstration of the power of metadata standardization, and to shed light on neuroanatomy in the human cortex, we developed a connectomics query engine which supports the analysis of the H01 dataset. Our application, called the H01 Community Discovery Portal, is currently deployed in the

TABLE 1 | Key annotation metadata definitions.

Uniform resource identifier (URI)	A link to specify where source data is located
Links	URIs to parent, child, sibling relationships
Data representation	The format used to represent a neuroanatomical entity (e.g., skeletons, meshes, or pixels)
Entity	An object with neuroscience significance (e.g., RAMON types: neurons, synapses, organelles); has properties
Property	An attribute and value such as weight, cell type, or layer
Community-defined entity	An entity with a community-adopted definition and a minimum set of required properties; can be extended
Community-defined property	A property with a community-adopted definition
User-defined entity	An entity without a community-adopted definition or minimum set of required properties; an entity defined by the user
User-defined property	A property without a community-adopted definition; a property defined by the user

Amazon Web Services cloud and follows the Representational State Transfer (REST) software architecture to ensure flexibility for storage of neuroanatomical metadata. The discovery portal consists of a Flask-based (Grinberg, 2018) web application, which serves as a user-friendly interface for researchers to explore and query the H01 dataset, a standards-supported H01 database, and a Flask-based Application Programming Interface (API) to enable easy access to the H01 dataset. We note that this is a simple example demonstrating the concepts outlined in this article and additional systems (e.g., neuPrint, CATMAID) might be used as robust alternatives with an appropriate schema.

The API is a web service consisting of eight RESTful web endpoints which retrieve and return H01 synapse, neuron, and layer data when a specific URL is accessed over Hypertext Transfer Protocol. The H01 Community Discovery Portal stores annotation metadata in a document-oriented, MongoDB (Chodorow, 2013) database.

4.2. Storing and Accessing the H01 Data

In the H01 database, we store nearly 27 GB of synapse, neuron, and layer properties along with their properties as MongoDB collections as described below:

- **Neuron Object:** Neuron ID (Integer), Volume (NumberLong), No. Outgoing Synapses (Integer), No. Incoming Synapses (Integer), No. Incoming Excitatory Synapses (Integer), No. Incoming Inhibitory Synapses (Integer), No. Dendrite Skeleton Nodes (Integer), No. Axon Skeleton Nodes (Integer), No. Dendritic Spines Skeleton Nodes (Integer), No. Cilia Skeleton Nodes (Integer), No. Axon Initial Segment Skeleton Nodes (Integer), No. Myelinated Axon Skeleton Nodes (Integer), Spinyness (Double), Layer (Integer), Neuron Type (Integer), Excitatory/Inhibitory Synapse Balance (Double)
- **Layer Object:** Layer Width (Integer)
- **Synapse Object:** Synapse ID (ObjectID), Synapse Type (Integer), Pre-synaptic site (Object), Post-synaptic partner (Object), Location (Integer), Bounding Box (Integer), Layer (Integer)

Each H01 synapse, neuron, and layer entity has an arbitrary amount of key-value properties which represent the object's metadata and attributes. Currently, each layer object has one attribute, each neuron object has 19 attributes, and each synapse object has seven attributes. Additionally, some attributes link to other entities with their own properties. For example, the synapse object has attributes, pre-synaptic site and post-synaptic partner, which contain sub-attributes, such as the associated neuron's id and class type. The document-oriented storage approach allows for H01 annotation attributes to be stored as arbitrary key-value pairs in which attributes can be easily added, queried, and indexed.This method served its primary purpose of demonstrating metadata standardization benefits, and we did not explore other database types.

4.3. Querying Data

We demonstrate a querying tool which performs standard queries relevant to a variety of connectomics research areas from the H01 dataset using the RAMON API. It is in the form of a web application with a user-friendly interface and provides a centralized location where users can easily explore the dataset individually through a personal query page (**Figure 1**), only accessible after the user is authenticated, as well as collaboratively through a "Popular Queries" page, accessible to all users.

The web application, located at metadata.bossdb.org, is also a Flask app which uses SQLite to store query and user information. It provides users with the ability to extract data via dropdown menus, enables visualization of cells and their synapses via Neuroglancer, supports reporting of potentially problematic annotations, compiles questions and answers from all users to generate a "Popular Queries" page, and even allows users to suggest new queries for integration into the app. At the moment, users can ask seven types of questions about the H01 dataset for a total of 119 questions shown below:

1. How many [Synapse Type] synapses are made in [Location]?
2. Which layer has the [Most/Fewest] [Synapse Type] synapses?
3. How wide is [Layer]?
4. What is the [Average/Total] length of neurons in [Location]?
5. Show me the neuron with the greatest number of [Synapse Type] synapses in [Location].
6. Show me a neuron with an $\frac{E}{E+I}$ value of [Value between 0 and 1] in [Location].
7. List all neurons with an $\frac{E}{E+I}$ value of [Greater/Less] than [Value between 0 and 1] in [Location].

where synapse types include all synapses, excitatory and inhibitory synapses, and those onto axon initial segments or dendrites, locations include the entire volume or any of the seven

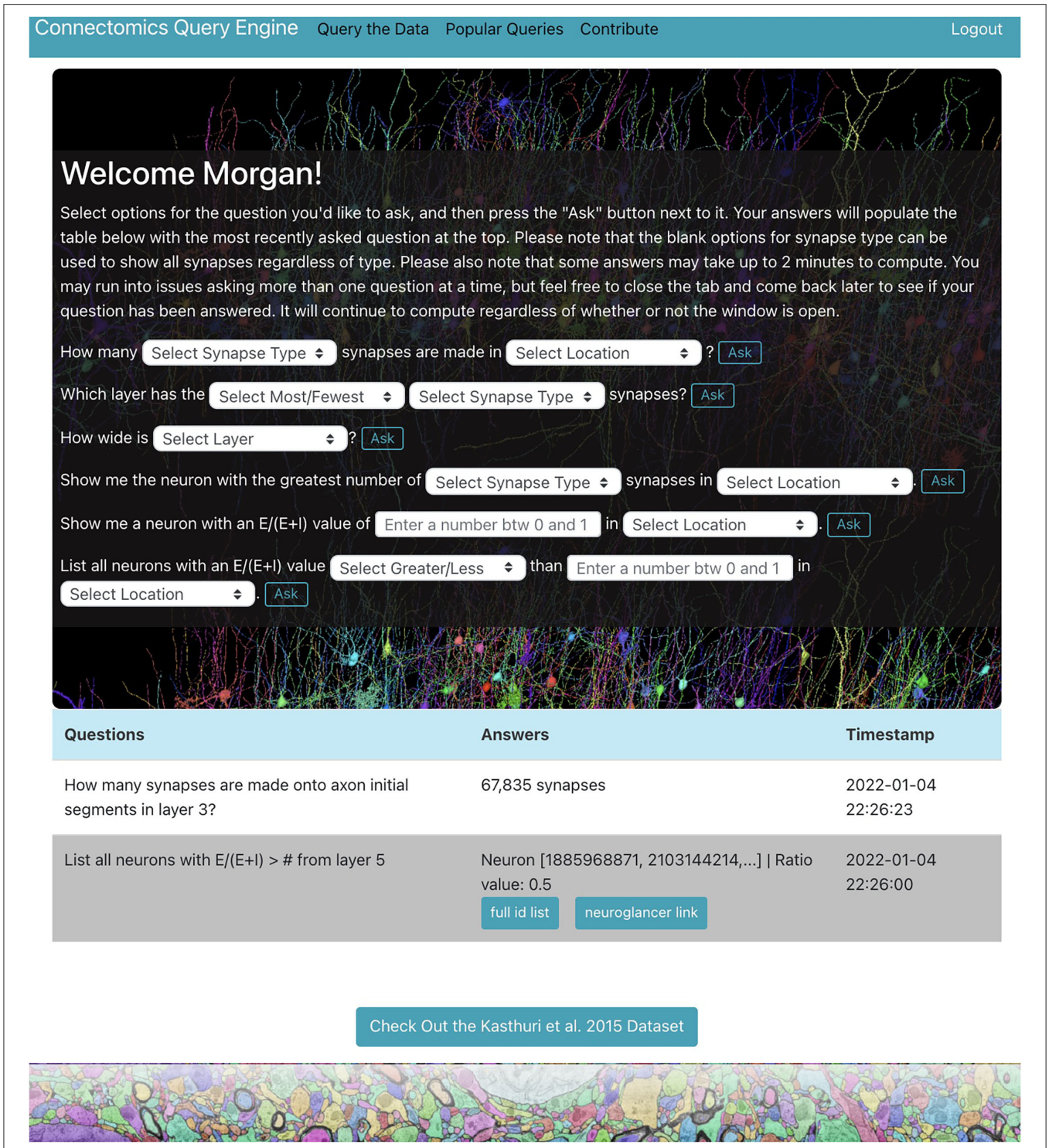


FIGURE 1 | H01 personal query page. Users can ask questions about the H01 dataset using dropdown menus and text boxes. Some answers are well-suited for visualization. The app uses Neuroglancer to display neurons related to such answers.

layers, and E and I are the number of incoming excitatory and inhibitory synapses, respectively.

This list of question types will continue to expand as functionality is added. We hope to continue to incorporate query types, especially those discussed in Section 3.

5. DISCUSSION

Nanoscale connectomics is an exciting field that has the potential to answer a wide range of questions in neuroscience and the potential to impact exciting and diverse application areas. The

number and size of EM and XRM datasets are growing, and with that growth comes an increased need for standardization of both imaging data and annotations. Work to standardize imaging data is underway, while standards to address annotation format and content variability are still emerging. As the field exists today, extracting relevant information about labeled entities requires both a significant computational background and a deep understanding of how that particular data is stored. Consequently, data access tools built upon annotation standards will increase accessibility and ease of access to these exciting datasets.

Standardization of metadata, while maintaining the flexible spirit embodied by an emerging field lays the groundwork for community-adopted standards which enable reproducible analysis as well as natural standards evolution over time. RAMON, RAMON API, and the H01 Community Discovery Portal demonstrate one example of how annotation standards and software tools can interact to support both collaborative and individual scientific discovery, but the possibilities are endless. The H01 Discovery Portal demonstrates how metadata standardization can push the field of connectomics toward solving potential applications by reducing redundant data processing code and encouraging data exploration and collaboration. All three of these tools have the potential to evolve to include additional queries, data sources, and annotation types.

Although it would be convenient to develop fully-automated pipelines to convert from lab-specific implementations to a common schema, due to the diversity in storage formats currently implemented, this will require future work. However, the process of understanding and translating important datasets has relatively low-resource requirements and can be simplified by focusing on the final, published data, which tends to be more standardized and common than intermediate products. The authors extended the portal to include a query page for Kasthuri et al. (2015) in addition to the H01 dataset. Though this data contained different entities, it existed in a tabular format similar to RAMON, which allowed for quick integration into the software stack.

The authors note, however, that the implementations of these tools may not be optimal as they were built primarily for standards demonstration purposes and thus serve as a proof of concept. For example, only MongoDB was considered for storing H01. In order to determine the best type of database to use for metadata storage, additional options such as DynamoDB, Google Cloud Firestore, Cassandra, and Azure Cosmos DB must be explored. Additionally, the web interface went through a small number of internal design cycles with particular emphasis on simple, clear, and intuitive querying. A more polished portal would necessitate an extensive design and feedback process. The authors hope to develop similar tools once standards are developed that allow for intuitive exploration of numerous datasets in a centralized location through expansion and integration of existing tools as well as development of new ones.

Further, we note that a top-down, universal specification of metadata standardsis unlikely to satisfy all stakeholders. For future work, we will ,therefore, seek a data-driven approach leveraging existing published data (Hider et al., 2019) and explicit community input. Standardization is often a balance between flexibility and usability, and we believe a fruitful path forward is to concentrate initially on published products.

Given the current limited accessibility of connectomics data, the patterns in brain networks may remain hidden behind these complex data, and scientific discovery could be limited. We look forward to building on these initial tools and formats through community engagement and feedback.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://h01-release.storage.googleapis.com/landing.html; https://bossdb.org/project/kasthuri2015.

AUTHOR CONTRIBUTIONS

MS and DM designed and developed the case study and validated the proposed standardization framework. EJ, BW, and WG-R proposed and defined the standards and perspectives outlined in this work, with input from the community. JL proposed the case study and provided neuroscience interpretation and design feedback. All authors contributed to the manuscript drafting and reviews and approved the submitted version of the manuscript.

FUNDING

Research reported in this publication was also supported by the National Institute of Mental Health of the National Institutes of Health under Award Numbers R24MH114799, R24MH114785, U24NS109102, and R01MH126684.

ACKNOWLEDGMENTS

We would like to thank the Johns Hopkins Applied Physics Laboratory ASPIRE program, especially students Jore Adegboyo and Kimberly Gordon for their contributions toward the H01 case study and building the related software tools. The completion of this project also could not have been possible without the guidance and expertise of the Johns Hopkins Applied Physics Laboratory BossDB team as well as the Cohort-based Integrated Research Community for Undergraduate Innovation and Trailblazing (CIRCUIT) program which provided training and coordination of students on this project. We also thank Viren Jain and his team at Google and members of the Lichtman Lab at Harvard University.

REFERENCES

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

Bassett, D. S., and Bullmore, E. (2006). Small-world brain networks. *Neuroscientist* 12, 51–523. doi: 10.1177/1073858406293182

Berger, D. R., Seung, H. S., and Lichtman, J. W. (2018). Vast (volume annotation and segmentation tool): efficient manual and semi-automatic labeling of large 3D image stacks. *Front. Neural Circuits* 12:88. doi: 10.3389/fncir.2018.00088

Boergens, K. M., Berning, M., Bocklisch, T., Bräunlein, D., Drawitsch, F., Frohnhofen, J., et al. (2017). Webknossos: efficient online 3D data annotation for connectomics. *Nat. Methods* 14, 691–694. doi: 10.1038/nmeth.4331

Chodorow, K. (2013). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. Sebastopol, CA: O'Reilly Media, Inc.

Clements, J., Dolafi, T., Umayam, L., Neubarth, N. L., Berg, S., Scheffer, L. K., et al. (2020). neuprint: analysis tools for EM connectomics. *bioRxiv*. doi: 10.1101/2020.01.16.909465

Consortium, M., Alexander Bae, J., Baptiste, M., Bodor, A. L., Brittain, D., et al. (2021). Functional connectomics spanning multiple areas of mouse visual cortex. *bioRxiv*. doi: 10.1101/2021.07.28.454025

Gray Roncal, W. R., Kleissas, D. M., Vogelstein, J. T., Manavalan, P., Lillaney, K., Pekala, M., et al. (2015). An automated images-to-graphs framework for high resolution connectomics. *Front. Neuroinformatics* 9:20. doi: 10.3389/fninf.2015.00020

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications With Python*. Sebastopol, CA: O'Reilly Media, Inc.

Hider, R., Kleissas, D. M., Pryor, D., Gion, T., Rodriguez, L., Matelsky, J., et al. (2019). The block object storage service (bossDB): a cloud-native approach for petascale neuroscience discovery. *bioRxiv*. 217745.

Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., et al. (2015). Saturated reconstruction of a volume of neocortex. *Cell* 162, 648–661. doi: 10.1016/j.cell.2015.06.054

LaGrow, T. J., Moore, M. G., Prasad, J. A., Davenport, M. A., and Dyer, E. L. (2018). "Approximating cellular densities from high-resolution neuroanatomical imaging data," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Honolulu, HI), 1–4. doi: 10.1109/EMBC.2018.8512220

Maitin-Shepard, J. (2020) *Neuroglancer*. Available online at: https://github.com/google/neuroglancer (accessed on May 1, 2020).

Matelsky, J. K., Reilly, E. P., Johnson, E. C., Stiso, J., Bassett, D. S., Wester, B. A., et al. (2021). Dotmotif: an open-source tool for connectome subgraph isomorphism search and graph queries. *Sci. Rep.* 11, 1–14. doi: 10.1038/s41598-021-91025-5

Plaza, S. M., Scheffer, L. K., and Chklovskii, D. B. (2014). Toward large-scale connectome reconstructions. *Curr. Opin. Neurobiol.* 25, 201–210. doi: 10.1016/j.conb.2014.01.019

Rübel, O., Tritt, A., Dichter, B., Braun, T., Cain, N., Clack, N., et al. (2019). *NWB: N 2.0: An Accessible Data Standard for Neurophysiology*. Cold Spring Harbor Laboratory. doi: 10.1101/523035

Saalfeld, S., Cardona, A., Hartenstein, V., and Tomančák, P. (2009). CATMAID: collaborative annotation toolkit for massive amounts of image data. *Bioinformatics* 25, 1984–1986. doi: 10.1093/bioinformatics/btp266

Schneider-Mizell, C. M., Bodor, A. L., Collman, F., Brittain, D., Bleckert, A. A., Dorkenwald, S., et al. (2020). Chandelier cell anatomy and function reveal a variably distributed but common signal. *bioRxiv*. doi: 10.1101/2020.03.31.018952

Shapson-Coe, A., Januszewski, M., Berger, D. R., Pope, A., Wu, Y., Blakely, T., et al. (2021). A connectomic study of a petascale fragment of human cerebral cortex. *bioRxiv*. doi: 10.1101/2021.05.29.446289

Sporns, O., Kötter, R., and Friston, K. J. (2004). Motifs in brain networks. *PLoS Biol.* 2:e369. doi: 10.1371/journal.pbio.0020369

Sunkin, S. M., Ng, L., Lau, C., Dolbeare, T., Gilbert, T. L., Thompson, C. L., et al. (2012). Allen brain atlas: an integrated spatio-temporal portal for exploring the central nervous system. *Nucl. Acids Res.* 41, D996–D1008. doi: 10.1093/nar/gks1042

Vogelstein, J. T., Perlman, E., Falk, B., Baden, A., Roncal, W. G., Chandrashekhar, V., et al. (2018). A community-developed open-source computational ecosystem for big neuro data. *Nat. Methods* 15, 846–847. doi: 10.1038/s41592-018-0181-1

Witvliet, D., Mulcahy, B., Mitchell, J. K., Meirovitch, Y., Berger, D. R., Wu, Y., et al. (2021). Connectomes across development reveal principles of brain maturation. *Nature* 596, 257–261. doi: 10.1038/s41586-021-03778-8

Xu, C. S., Januszewski, M., Lu, Z., Takemura, S.-Y., Hayworth, K. J., Huang, G., et al. (2020). A connectome of the adult drosophila central brain. *BioRxiv*. doi: 10.1101/2020.01.21.911859

Yatsenko, D., Reimer, J., Ecker, A. S., Walker, E. Y., Sinz, F., Berens, P., et al. (2015). Datajoint: managing big scientific data using MATLAB or python. *BioRxiv*. 031658. doi: 10.1101/031658

Author Disclaimer: The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Conflict of Interest: 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.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership