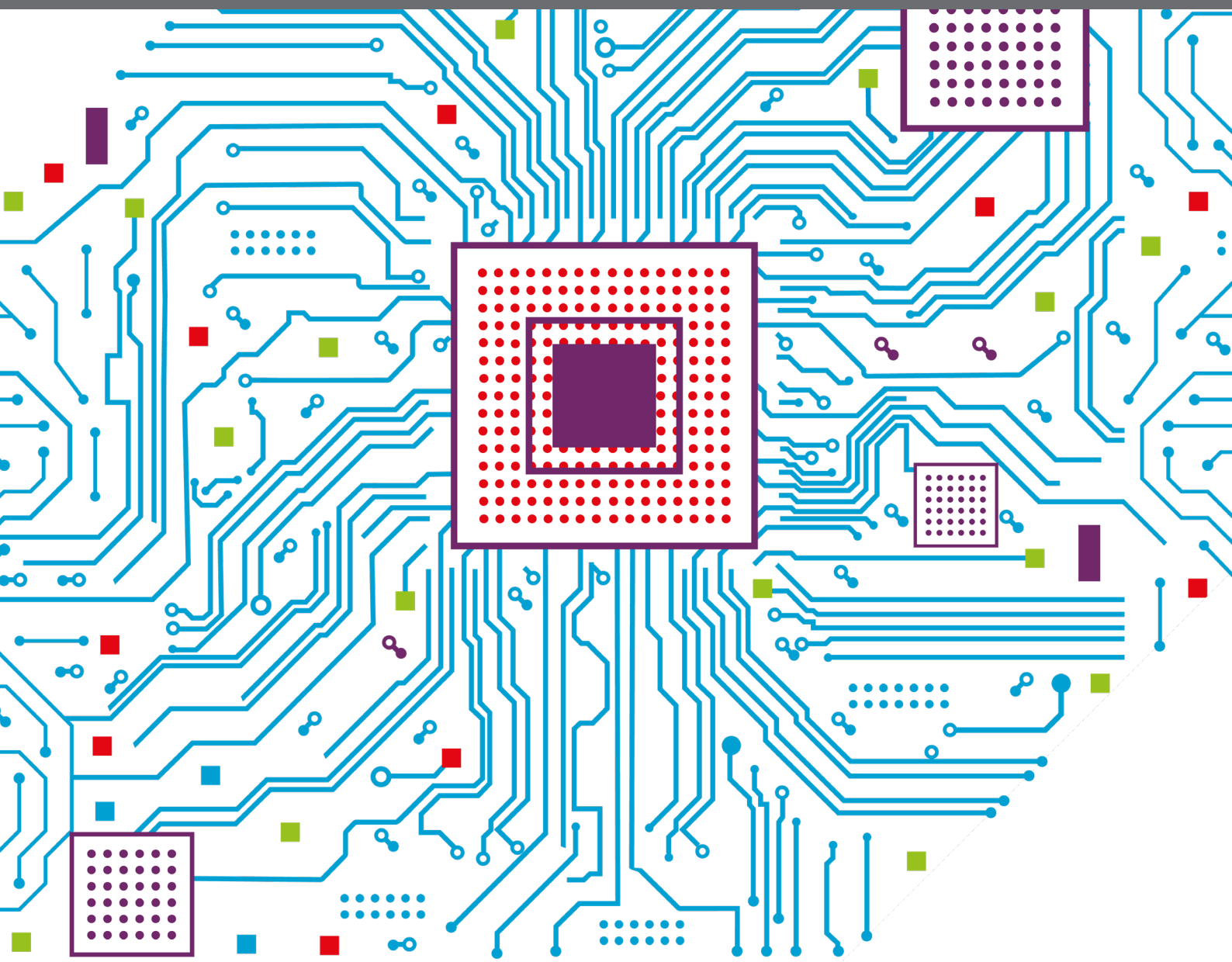
METHODS AND TOOLS FOR BIOIMAGE ANALYSIS

EDITED BY: Florian Levet, Florian Jug and Virginie Uhlmann
PUBLISHED IN: Frontiers in Computer Science



frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88976-406-8
DOI 10.3389/978-2-88976-406-8

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

METHODS AND TOOLS FOR BIOIMAGE ANALYSIS

Topic Editors:
Florian Levet, UMR5297 Institut Interdisciplinaire de Neurosciences (IINS), France
Florian Jug, Human Technopole, Italy
Virginie Uhlmann, European Bioinformatics Institute (EMBL-EBI), United Kingdom

Citation: Levet, F., Jug, F., Uhlmann, V., eds. (2022). Methods and Tools for Bioimage Analysis. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88976-406-8

Table of Contents

frontiers | Frontiers in Computer Science

EDITORIAL
published: 30 May 2022
doi: 10.3389/fcomp.2022.931939

Editorial: Methods and Tools for Bioimage Analysis

Florian Levet[1,2]*, *Virginie Uhlmann*[3]* and *Florian Jug*[4]*

[1] Univ. Bordeaux, CNRS, Interdisciplinary Institute for Neuroscience, IINS, UMR 5297, Bordeaux, France, [2] Univ. Bordeaux, CNRS, INSERM, Bordeaux Imaging Center, BIC, UAR3420, US 4, Bordeaux, France, [3] European Bioinformatics Institute, European Molecular Biology Laboratory (EMBL), Cambridge, United Kingdom, [4] Fondazione Human Technopole, Milan, Italy

Keywords: bioimage analysis, methods, tools, software platform, accessibility

Editorial on the Research Topic

Methods and Tools for Bioimage Analysis

A core purpose of biological imaging data is the quantification of complex phenomena through bioimage analysis. Recent advances in our ability to observe living systems, enabling life scientists to monitor the dynamics of biological phenomena across spatiotemporal scales and at high resolution, have led to a much increased demand for methods and tools to analyze microscopy data. In many scientific projects in the life sciences, the bottleneck has now shifted from not having the technology to image interesting phenomena to not being able to extract information from the unfathomable amounts of image data acquired during said imaging experiments.

Historically, Bioimage Analysis operates in between the disciplines of computer science, physics, microscopy, medicine, and biology, presenting a rather unique set of challenges on various levels. With growing demands, the community grew, and saw the emergence of national and international networks such as NEUBIAS (http://eubias.org/NEUBIAS/), COBA (https://openbioimageanalysis.org/), or BINA (https://www.bioimagingnorthamerica.org/), all aiming at connecting bioimage analysis researchers, helping them to better align their interest, and ultimately be more efficient and less redundant. This has also helped this community to better understand what its strengths are, and where bioimage analysis methods are still missing.

An interesting and quite unique aspect of our work as Bioimage Analysts is that we need to develop not only methods but also tools—both tasks being exquisitely time-consuming and requiring very different skills. Our "users," on the other hand, are experts in the specific biological systems they study and should ideally not require beyond average computational skills. Hence, even the most powerful new computational method is of little to no use if not cast into accessible and usable software tools.

We present a Research Topic of useful new methods, all cast into accessible software tools that enable life-scientists to analyze their microscopy image data.

In this Research Topic we collected work that presents methods needed to analyze microscopy image data, all of which are cast into accessible software tools. We cover a wide range of pressing bioimage analysis tasks, with individual contributions addressing either specific or common analysis problems, in ways that might focus on describing novel methodological ideas or on making known methods and approaches available in efficient and user-friendly ways. All presented work has one thing in common: the promise to enable the analysis of scientific image data in the hands of our users.

Four contributions have a particularly interesting focus on novel methodological approaches, and are essential for pushing forward robustness and automation in the bioimage analysis field. Ulicna et al. develop a hybrid deep learning and Bayesian cell tracking approach to reconstruct

OPEN ACCESS

Edited and reviewed by:
Marcello Pelillo,
Ca' Foscari University of Venice, Italy

***Correspondence:**
Florian Jug
florian.jug@fht.org
Virginie Uhlmann
uhlmann@ebi.ac.uk
Florian Levet
florian.levet@u-bordeaux.fr

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *29 April 2022*
Accepted: *16 May 2022*
Published: *30 May 2022*

Citation:
Levet F, Uhlmann V and Jug F (2022) Editorial: Methods and Tools for Bioimage Analysis. Front. Comput. Sci. 4:931939. doi: 10.3389/fcomp.2022.931939

lineage trees from live-cell microscopy data, offering their novel approach to users in an open Python-based software tool. The work by Haase demonstrates the utility of defining image filtering operations directly on a grid of biologically-relevant objects such as cells instead of image pixels, thereby rendering these filtering operations efficient enough for real-time applications even on sizable image data. In Matskevych et al., the authors observe that the power of deep learning based segmentations and the simplicity of a random forest based pixel classification can be combined in an intriguing new transfer learning idea. Finally, Klatzow et al. combine several state-of-the-art concepts for surface matching into a fully automated shape correspondence pipeline, demonstrated to efficiently streamline the landmark-free alignment of complex 3D objects for downstream shape analysis.

Other work we present excels in making potentially widely applicable ideas and methods better accessible to a wide range of users and/or applicable to truly large datasets. While biological image data is filled with fibrillar structures of various kinds, quantifying their orientation is often a critical step. Marcotti et al. present a method, with open sources in Matlab and Python, allowing users to efficiently and accurately quantify both local and global alignments by fibrillar structure using a Fourier-based alignment strategy. In Chiaruttini et al., the authors introduce Warpy, a pipeline for solving another pressing alignment problem, namely the alignment of whole slides in histopathology. Their solution combined state of the art software platforms (elastix, Fiji and its BigWarp plugin, and QuPath), enabling users to perform semi-automated non-linear registration of their data through robust user interfaces. In Arzt et al., the authors present Labkit, a random forest based trainable pixel classifier capable of segmenting truly large volumetric image and time-series data. Labkit offers an intuitive user interface, and ows its efficiency to BigDataViewer and imglib2, two essential parts of the modern Fiji ecosystem (https://fiji.sc/). While Fiji still is a pillar of many tools and workflows, napari (https://napari.org/) has recently begun to take the bioimage analysis field by storm. It allows users to visualize and manipulate their images in a Python-based environment, with n-dimensionality being one of napari's core principles. D'Antuono and Pisignano present ZELDA, a plugin for napari, designed to enable users to build customizable bioimage analysis workflows, demonstrated in the context of 3D. Finally, Ritchie et al. present Tonga, a new software platform that gives emphasis to ease-of-use and user-friendliness. Tonga's wizard feature suggests suitable methods to its users, depending on the image and the task at hand, with a special focus on detection and segmentation of nuclei.

The final group of contributions enable specific bioimage quantification investigations. By definition, these pipelines and tools are not intended to be generally applicable, but without them, the research they support would become impossibly time-consuming to conduct, and insights hard to gain. Rodriguez et al. present a ML based computer vision approach to quantify the foraging behavior of bees at the entrance of their hive using deep learning for detecting bees and then tracking them while also estimating their pose. In Rahm et al., the authors introduce a mean-squared displacement analysis to classify single-molecule tracklets into immobile, confined diffusing, and freely diffusing states. Their system can also detect transitions between these modes, allowing them to better understand the molecular system they study. Last but not least, Schmied et al. present SynActJ, an ImageJ plugin combined with a R Shiny app designed for the automatic detection and analysis of synaptic activity in time-lapse movies. SynActJ allows end-to-end analysis, from filter-based synapse segmentation, to the full analysis of all previously segmented traces within a user-friendly tool.

In summary, this Research Topic provides a collection of bioimage analysis methods and open software tools, developed with the needs of our community in mind. We strongly believe that the work presented in this Research Topic demonstrates how broad our field is and how important it is to conduct research that contributes with not only novel methods to pressing analysis problems, but also with sound solutions that are open, available, and applicable by others to their own data. If our community keeps pushing these virtues, progress in the life sciences will be making progress in bigger and bigger strides.

AUTHOR CONTRIBUTIONS

FJ, VU, and FL conceived the Research Topic and wrote the manuscript. All authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

We wanted to thank all the contributors to this Research Topic for their insightful and interesting manuscripts.

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 Computer Science

ORIGINAL RESEARCH
published: 14 October 2021
doi: 10.3389/fcomp.2021.745831

A Workflow for Rapid Unbiased Quantification of Fibrillar Feature Alignment in Biological Images

Stefania Marcotti[1], Deandra Belo de Freitas[1], Lee D Troughton[2], Fiona N Kenny[1], Tanya J Shaw[3], Brian M Stramer[1] and Patrick W Oakes[2]**

[1]Randall Centre for Cell and Molecular Biophysics, King's College London, London, United Kingdom, [2]Department of Cell and Molecular Physiology, Stritch School of Medicine, Loyola University Chicago, Maywood, IL, United States, [3]Centre for Inflammation Biology & Cancer Immunology, King's College London, London, United Kingdom

Measuring the organization of the cellular cytoskeleton and the surrounding extracellular matrix (ECM) is currently of wide interest as changes in both local and global alignment can highlight alterations in cellular functions and material properties of the extracellular environment. Different approaches have been developed to quantify these structures, typically based on fiber segmentation or on matrix representation and transformation of the image, each with its own advantages and disadvantages. Here we present *AFT – Alignment by Fourier Transform*, a workflow to quantify the alignment of fibrillar features in microscopy images exploiting 2D Fast Fourier Transforms (FFT). Using pre-existing datasets of cell and ECM images, we demonstrate our approach and compare and contrast this workflow with two other well-known ImageJ algorithms to quantify image feature alignment. These comparisons reveal that *AFT* has a number of advantages due to its grid-based FFT approach. 1) Flexibility in defining the window and neighborhood sizes allows for performing a parameter search to determine an optimal length scale to carry out alignment metrics. This approach can thus easily accommodate different image resolutions and biological systems. 2) The length scale of decay in alignment can be extracted by comparing neighborhood sizes, revealing the overall distance that features remain anisotropic. 3) The approach is ambivalent to the signal source, thus making it applicable for a wide range of imaging modalities and is dependent on fewer input parameters than segmentation methods. 4) Finally, compared to segmentation methods, this algorithm is computationally inexpensive, as high-resolution images can be evaluated in less than a second on a standard desktop computer. This makes it feasible to screen numerous experimental perturbations or examine large images over long length scales. Implementation is made available in both MATLAB and Python for wider accessibility, with example datasets for single images and batch processing. Additionally, we include an approach to automatically search parameters for optimum window and neighborhood sizes, as well as to measure the decay in alignment over progressively increasing length scales.

Keywords: alignment, fast Fourier transform (FFT), cytoskeleton, extracellular matrix (ECM), fibers, anisotropy

OPEN ACCESS

Edited by:
Florian Levet, UMR5297 Institut Interdisciplinaire de Neurosciences (IINS), France

Reviewed by:
David J Barry, Francis Crick Institute, United Kingdom
Anna Helene Klemm, Uppsala University, Sweden

****Correspondence:***
Stefania Marcotti
stefania.marcotti@kcl.ac.uk
Patrick W Oakes
poakes@luc.edu

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *22 July 2021*
Accepted: *21 September 2021*
Published: *14 October 2021*

Citation:
Marcotti S, Belo de Freitas D, Troughton LD, Kenny FN, Shaw TJ, Stramer BM and Oakes PW (2021) A Workflow for Rapid Unbiased Quantification of Fibrillar Feature Alignment in Biological Images. Front. Comput. Sci. 3:745831. doi: 10.3389/fcomp.2021.745831

1 INTRODUCTION

Measuring the anisotropy of features in biological images is of increasing interest as the degree of alignment can inform on both the underlying cellular behaviors and material properties of the sample. For example, different cell types have unique emergent capacities to align in culture, controlled by cellular packing and geometry (Duclos et al., 2014, Duclos et al., 2017). When cells are exposed to external forces such as cyclical stretch, they tend to align their cytoskeleton perpendicular to the axis of stretch (Standley et al., 2002; Yoshigi et al., 2005; Livne et al., 2014). Subcellular cytoskeletal networks can also spontaneously organize in response to the stresses of their environment (Gupta et al., 2015, 2019) or changes in biochemical signalling (Ridley and Hall, 1992). Similar changes in cytoskeletal architecture can be seen in reconstituted protein systems (Falzone et al., 2013; Linsmeier et al., 2016). Additionally, the extracellular matrix (ECM) also has an inherent capacity to align in different tissues or pathologies, such as cancer (Ouellette et al., 2021) or tissue fibrosis (Park et al., 2020; Mascharak et al., 2021), which is thought to alter the ECM network mechanical properties. These examples illustrate the variety of environments where alignment of features reveals important biological properties. As such, it is necessary to develop approaches to efficiently quantify anisotropy of features across a range of length scales, from subcellular organization to tissue level alignment.

A number of different approaches have been developed to quantify the alignment of image features. These methods can be categorized based on the type of algorithm used to highlight features and subsequently quantify anisotropy. Fiber segmentation tools, such as the Fiji Ridge Detection plugin (Lindeberg, 1998), the curvelet-based CurveAlign/CT-FIRE suite (Bredfeldt JS. et al., 2014; Bredfeldt J. S. et al., 2014; Liu et al., 2017, 2020), and Filament Sensor (Eltzner et al., 2015), provide individual fiber information. Tools based on representation/transformation of the image, such as the Fiji plugins OrientationJ (Rezakhaniha et al., 2012) and FibrilTool (Boudaoud et al., 2014) or the CytoSpectre suite (Kartasalo et al., 2015), supply overall fiber alignment information. Hybrid tools, such as TWOMBLI, which exploit a combination of approaches (i.e., fiber segmentation followed by matrix representation of the image) have also been recently made available (Wershof et al., 2021).

Many image transformation algorithms rely on Fourier transformation of the image and exploit its frequency space representation to obtain alignment information (Chaudhuri et al., 1987; Pourdeyhimi et al., 1997; Marquez, 2006; Sander and Barocas, 2009; Goldyn et al., 2010; Kartasalo et al., 2015; Clemons et al., 2018). These algorithms tend to be computationally efficient (Sander and Barocas, 2009) and insensitive to the image background noise (Liu et al., 2020), which is desirable for rapidly processing large volumes of experimental data (Püspöki et al., 2016). Another desirable trait for an alignment detection tool is its flexibility in terms of size and location of alignment patterns (Püspöki et al., 2016), which is handled well by the only open-source software available in this category (CytoSpectre (Kartasalo et al., 2015)) due to a power spectrum analysis of the image's discrete Fourier transform.

One feature missing from the available tools in the literature is an analysis of the alignment length scale. Depending on the scientific question of interest, one may want to measure alignment over a small region of the image (e.g., local cytoskeletal features inside cells) or over a broad length scale (e.g., ECM organization in whole tissues). Additionally, when comparing experimental datasets, it is important to understand the precise length scale over which the observed alignment is significant and determine the spatial decay of the anisotropy, which will allow for better interpretation and analysis of results (e.g., is the alignment of a region of interest a subcellular or supracellular phenomenon). Here we explain the implementation of an open-source alignment quantification algorithm, *AFT – Alignment by Fourier Transform*, that can be run on a variety of biological images and has a number of advantages over pre-existing approaches. This FFT-based quantification is rapid and computationally efficient, flexible to different types of microscopy images and samples, easy to use with a low number of parameters, and importantly allows for a user-defined length scale over which to measure feature anisotropy.

2 MATERIALS AND METHODS

2.1 Biological Images

Sample datasets of microscopy images were used to illustrate the analysis and are provided as Supplementary Material. In particular, cultured fibroblasts transfected with GFP-actin, fibroblasts fixed and stained for filamentous actin (F-actin, phalloidin), fixed cell-derived matrices immunostained for actin (before decellularization) and fibronectin (after decellularization), and second harmonic generation imaging of tissue samples were utilized. Fixed actin images were acquired at different magnifications using either single cells or cells at confluence, to allow for evaluation of the length scale of alignment. Y-27632 treated cells were incubated in the indicated concentration of the drug for 30 min and then fixed and stained with phalloidin.

Fixed samples were imaged using a Zeiss LSM 880 equipped with a 40x NA 1.3 Plan-Apochromat oil objective or 63x NA 1.4 Plan-Apochromat oil objective. Decellularized matrices were imaged using a Zeiss LSM 880 equipped with 63x NA 1.4 Plan-Apochromat oil objective. For second harmonic generation imaging, tissue sections were imaged using a Zeiss LSM 7MP equipped with a 20x NA 1.0 water immersion objective. Y-27632 treated cells were imaged on a 3i Marianas Imaging System consisting of a Zeiss Axio Observer 7 inverted microscope attached to a Yokogawa W1 Confocal Spinning Disk using a 63x NA 1.4 Plan-Apochromat oil objective.

2.2 Synthetic Data

Images of fibrillar features were simulated in MATLAB (Mathworks, v2018b) to create sets of synthetic data with specific length scales. To this aim, squared units with a

pre-defined size were randomly distributed in the field of view. Each unit was filled with parallel lines with 5 *px* spacing, and then rotated with a random angle between 0° and 180°. Three 10-image sets were generated, with unit sizes equal to 100, 200, or 400 *px*.

2.3 Pre-Processing

Image pre-processing is not required for this analysis, but can be used to additionally highlight the fibrillar features. In the context of this work, pre-processing was performed only for the cell-derived matrix images, as their surface is often not flat, resulting in uneven signal across the field of view. To account for this, contrast was enhanced in Fiji (0.35% saturated pixels) and subsequently a local contrast adjustment was performed (CLAHE plugin, default parameters). All images shown are maximum projections of Z-stack acquisitions, except for the live cell images which represent a single confocal slice.

2.4 Comparison With Other Available Alignment Tools

The performance of the algorithm presented in this work was compared with OrientationJ (Rezakhaniha et al., 2012) and TWOMBLI (Wershof et al., 2021), by using two 10-image samples of fibronectin-stained cell-derived matrices that were either isotropic or anisotropic. To access the alignment vector field for OrientationJ, a Fiji macro was designed to batch run the OrientationJ Vector Field plugin. This operation was either performed on its own or preceded by fiber segmentation by the Ridge Detection plugin (Lindeberg, 1998), this second approach mimicking the TWOMBLI workflow (hybrid method). To obtain a comparable spatial resolution of the vector field, a window of 100 *px* with 50% overlap was used to run the FFT analysis, and a matching local window σ of 100 *px* with a grid size of 50 was employed in OrientationJ. The Ridge Detection parameters were set to: *line width* = 20 *px*, *high contrast* = 120, *low contrast* = 0, *sigma* = 6.27, *lower threshold* = 0, *upper threshold* = 0.17, *minimum line length* = 10 *px*. These parameters were selected to match the ones assigned by the TWOMBLI macro when calling the Ridge Detection plugin (see below for the user input parameters that were chosen in TWOMBLI). From the acquired vector fields, an order parameter over neighborhoods of 5*x* vectors was calculated and compared for the three approaches.

A second comparison was based on coherency, the metric traditionally used for global alignment output by both OrientationJ and TWOMBLI. Coherency of the FFT was measured on windows of 100 *px* with 50% overlap and averaged to obtain a median value representing the global alignment score for each image. To obtain the coherency score from OrientationJ, a macro was designed to batch process the images using the OrientationJ Dominant Direction plugin. TWOMBLI was run on the sample data as per the developer instructions. A parameter optimization step was performed on test images and the obtained parameter file was used for batch processing (*line width* = 20 *px*, *curvature window* = 150 *px*, *branch length* = 10).

Computational time was measured on a 3.6 GHz Quad-Core Intel Core i7 machine with 32 GB memory, with suitable functions in MATLAB and ImageJ macro language. Only the effective analysis time for a 10-image sample (1 MB per image) was taken into account, excluding the input and parameter upload. Timings for OrientationJ refer to the designed macro, no manual handling of data was involved.

2.5 Statistical Analysis

Statistical analysis was performed in Prism (Graphpad, v9). Mann-Whitney two-tailed tests were used to compare metrics on isotropic vs. anisotropic samples. Significance is reported in figure panels as follows: '****' for *p*-values lower than 0.0001, '***' lower than 0.001, '**' lower than 0.01, '*' lower than 0.05, 'ns' otherwise.

2.6 Implementations

This approach builds upon previous work employed in (Aratyn-Schaus et al., 2011; Cetera et al., 2014; Fernandes et al., 2020) and adds features such as periodic decomposition to improve the accuracy of the FFT angle determination along with the ability to search for the optimal length scale for comparison. The current implementation is available at https://github.com/OakesLab/AFT-Alignment_by_Fourier_Transform, both in MATLAB and Python languages. The MATLAB version is provided with a simple user interface for inputting parameters, while the Python version is presented as a set of Jupyter notebooks. The code is divided into two separate suites. The first suite can be used to perform the alignment quantification on a sample of images containing fibrillar structures. The second suite can be used to run a search to optimise the analysis parameters, with the aim of maximizing differences between two samples, i.e., locating the length scale for which the difference in alignment between two samples is greatest. Further code documentation is provided in the repository.

3 RESULTS

3.1 Measuring Alignment

The algorithm is inspired by the general approach of Particle Image Velocimetry (Willert and Gharib, 1991), where the image is broken down into a series of windows that are analyzed independently to create a vector field that represents the entire image. Windows are analyzed in frequency space to reveal information about both the fibrillar structure and local alignment as illustrated in **Figure 1** using filamentous actin images. If the image contains aligned features in the real space, the corresponding FFT in the frequency domain will be asymmetrically skewed, with the direction of skew orthogonal to the original feature orientation (**Figure 1A,B**). This process can then be repeated on each successive window across the image, resulting in a vector field that represents the local alignment in the image. A detailed protocol for determining the local alignment of each window and the order parameter of global alignment follows.

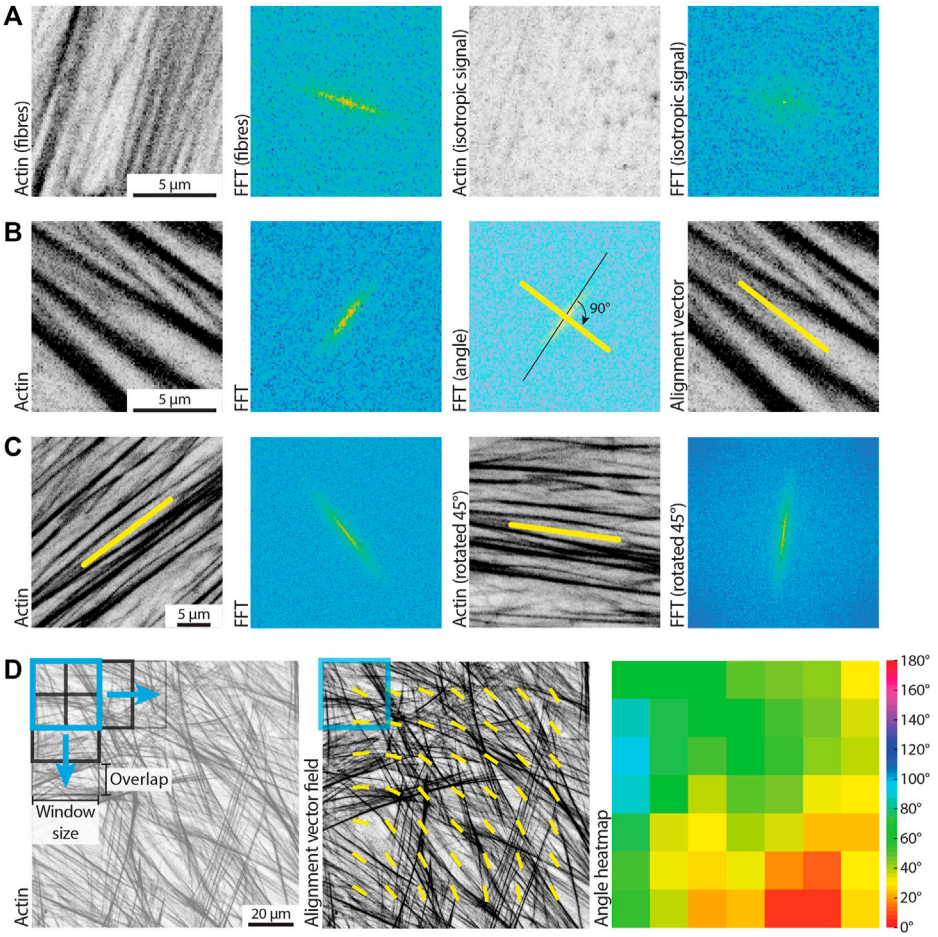


FIGURE 1 | Using FFTs to measure local alignment of biological images. **(A)** Small windows of a microscopic image are considered, containing either aligned filamentous actin fibers, or isotropic signal. The 2D Fast Fourier Transform (FFT) shows an elongated (i.e., skewed) shape if the image in the real space contains aligned fibrillar features, and displays a more round shape otherwise. Scale bar, 5 *µm*. **(B)** A small window of a microscopic image containing aligned filamentous actin fibers is considered. Its FFT shows a predominant skewness (black line) of an angle that is orthogonal to the direction of the fibers in the real space image. The obtained alignment vector (yellow line) can be overlaid to the original image to highlight the measured fiber orientation. Scale bar, 5 *µm*. **(C)** The alignment vector is calculated for an image containing aligned actin fibers by exploiting the characteristics of the FFT. The same image, rotated by 45°, is analyzed, to demonstrate robustness of the method to rotation. Scale bar, 5 *µm*. **(D)** A square representing the window size for analysis (250 *px*, light blue) is overlaid on a filamentous actin image. The black squares represent subsequent instances of such a window with the defined overlap, as the image is scanned during the analysis to calculate local alignment vectors (yellow). The heatmap shows a different representation of the alignment vector field, with a wrapped colour scale ranging from 0° to 180°. Scale bar, 20 *µm*.

3.1.1 Local Alignment

Square windows of *n x n* pixels are chosen from the original real space image and a 2D FFT is performed. Typically, *n* is chosen as odd to ensure that the zero-order frequency component, which is always the greatest in magnitude, is situated at the centre of the FFT. FFTs of non-periodic signals are often plagued by strong horizontal and vertical components, appearing as a cross, due to the mismatch of intensity in the images at the edges. To avoid this effect, we break down the image into its smooth and periodic components following the approach of Moisan (2011). We use the periodic component of this decomposition to take the FFT and then take its norm to only deal with real numbers. The resulting image is then multiplied by a mask of diameter *n*/2 to capture all the relevant high frequency components and to ensure the sample is symmetric. The central image moments of the masked FFT are then calculated as

$$\mu'_{ij} = \frac{\sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j I(x,y)}{\sum_{xy} I(x,y)}, \tag{1}$$

where *x* and *y* represent the position and *I* is the magnitude of the FFT norm. We next construct the covariance matrix of the image as

$$cov[I(x,y)] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix}, \tag{2}$$

which has eigenvalues of

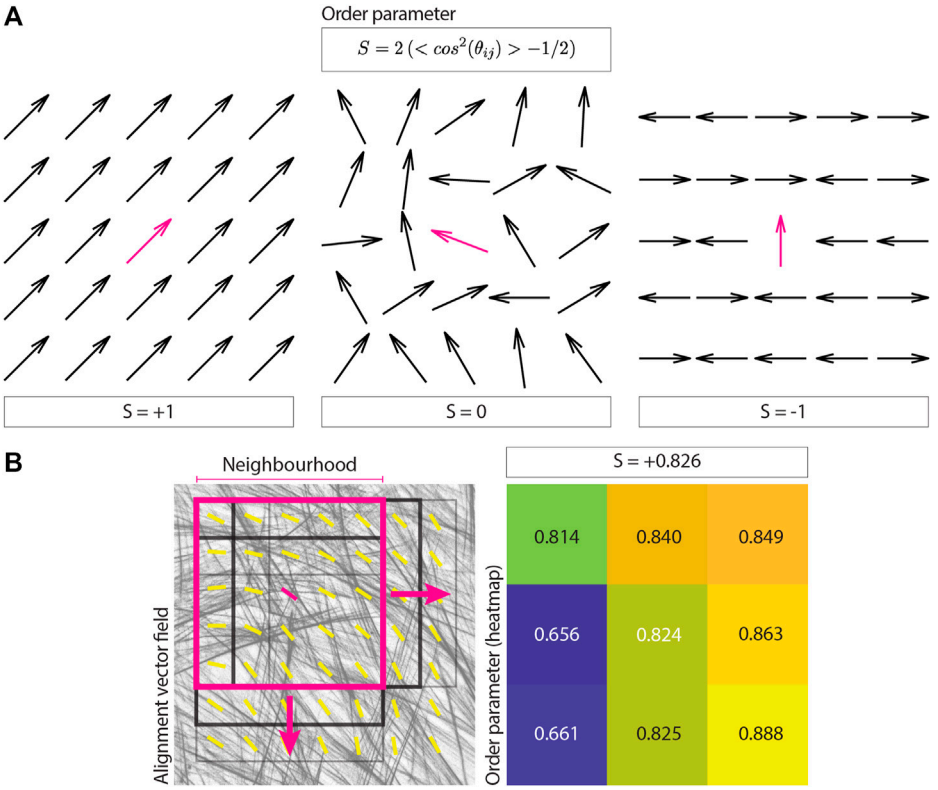


FIGURE 2 | Calculating an image order parameter. **(A)** The equation for the order parameter used to assign an alignment score for each neighborhood is shown, together with examples. The order parameter takes a value of 1 for complete alignment, 0 for random alignment, and −1 for orthogonal alignment. **(B)** A square representing a neighborhood of 5*x* vectors (magenta, neighborhood radius of 2*x* vectors around the central reference) is overlaid on an actin image and its corresponding alignment vector field. The black squares represent subsequent instances of such neighborhood, as the vector field is scanned during the analysis to obtain local alignment scores. The resulting order parameters for each neighborhood are reported in the heatmap, together with their median value, representing the output of the analysis.

$$\lambda_i = \frac{\mu'_{20} + \mu'_{02}}{2} \pm \frac{\sqrt{4\mu'^2_{11} + (\mu'_{20} - \mu'_{02})^2}}{2}. \quad (3)$$

The orientation of the FFT can thus be calculated by

$$\theta = \frac{1}{2} arctan\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right), \quad (4)$$

and the eccentricity, a measure of how oblong the FFT is, as

$$e = \sqrt{1 - \frac{\lambda_2}{\lambda_1}}. \quad (5)$$

The orientation of the features in the real image is orthogonal to the orientation of the FFT, and thus we apply a 90°rotation (**Figure 1B**). Orientation vectors do not have a polarity, and thus angles of 0° or 180° are considered equivalent. Therefore, all angles are mapped to a range of 0°–180°(**Figure 1D**). This methodology is robust to rotational transformations and does not suffer from the inherited bias of FFT for vertical and horizontal components (**Figure 1C**).

3.1.2 Order Parameter Calculation

Both the size of the window and the degree of overlap between the windows is customizable (**Figure 1D**). To measure how aligned fibrillar features are within adjacent windows, we define an order parameter by correlating the directionality of vectors within the neighborhood of customizable size. Specifically, the order parameter can be calculated as

$$S = 2\left(< cos^2\theta_{ij} > - \frac{1}{2}\right), \quad (6)$$

with θ_{ij} representing the angle between the orientation of a central reference vector and its neighbors (**Figure 2A**). The order parameter can take values between −1 and 1 (**Figure 2A**), with 1 representing perfect alignment (i.e., all vectors in the neighborhood have the same orientation as the reference vector), 0 representing random orientation, and −1 representing opposite alignment (i.e., all vectors in the neighborhood are pointing in the orthogonal direction compared to the reference vector). While a number of different formulations of the order parameter have been used previously, in the context of the current analysis, the chosen order parameter normally ranges between 0 (random) and 1 (perfectly aligned), as fiber polarity is not taken into account and the neighborhood area is kept relatively local (**Figure 2B**). Similar results were obtained for real and simulated data (and **Figure 3**).

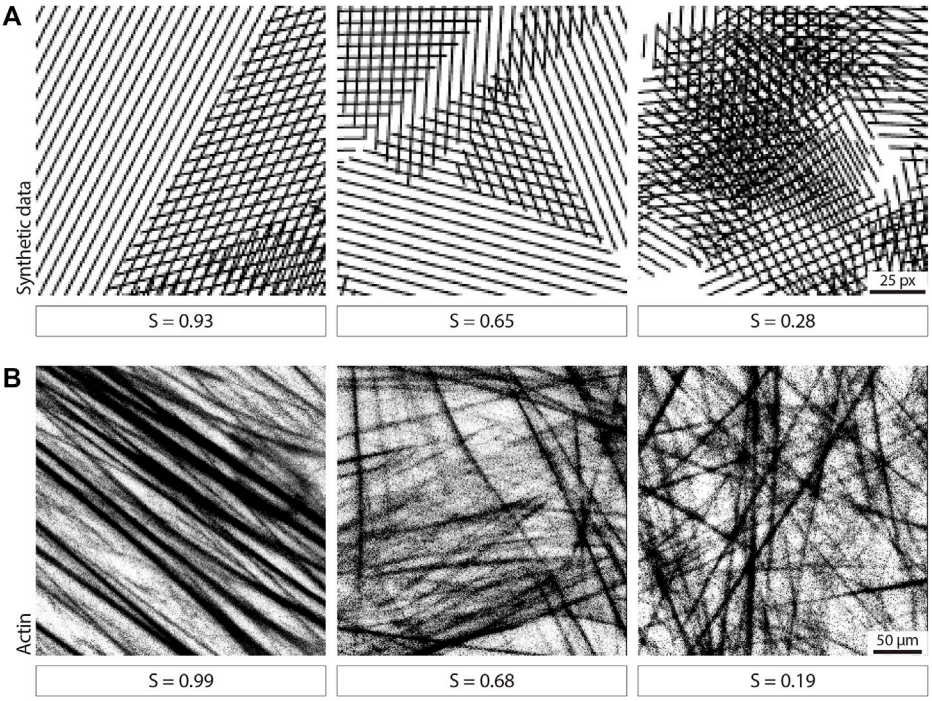


FIGURE 3 | Examples of image neighborhoods and corresponding order parameter. **(A)** Examples of neighborhoods in simulated data displaying different degrees of alignment are shown together with the calculated order parameter (window size 35 *px*). Scale bar, 25 *px*. **(B)** Examples of neighborhoods in actin images displaying different degrees of alignment are shown together with the calculated order parameter (window size 100 μm). Scale bar, 50 μm.

To calculate an overall alignment score for each image, we define a neighborhood size. The previously calculated vector field is then split into all possible overlapping neighborhoods of such size. The order parameter is subsequently calculated for each neighborhood, and the overall order parameter for the image is defined as the median value (**Figure 2B**).

3.2 Investigating the Optimal Alignment Length Scale Between Two Samples

When comparing two samples displaying different degrees of alignment, it may be of interest to evaluate the length scale for which this difference is greatest. The approach described above allows for this, as it is possible to evaluate the order parameter using a range of window and neighborhood sizes which correspond to different length scales (i.e., each permutation of window and neighborhood size define a specific length scale). By comparing the obtained alignment scores between the two experimental conditions for the different parameter permutations, it is possible to investigate the precise length scale at which the difference between the two samples is most pronounced. The comparison can be carried out by looking at the order parameter difference between the two samples or by running statistical tests and analyzing the resulting *p*-values.

We tested this parameter search approach on actin images of cultured fibroblasts with different degree of isotropy (**Figure 4A**, 10 images for each sample), for window sizes ranging from 25 to 325 *px* and neighborhood radii ranging from 1*x* to 38*x* vectors (**Figures 4B,C**). We performed the comparison in alignment scores for each window/neighborhood size permutation by quantifying the difference between the sample order parameters (**Figure 4B**) or by calculating the *p*-value of a Mann-Whitney statistical test between the two populations (**Figure 4C**).

It should be noted that small windows paired with small neighborhoods display noisy output (**Figures 4B,C**), as the corresponding FFT for such small windows will tend towards low eccentricity values (i.e., increasing the likelihood of spurious vectors, **Figure 1A**). Interestingly, when comparing two samples with known differences in alignment (**Figure 4A**), it can be observed how the difference in order parameter between the two samples displays a peak when keeping the window size constant and increasing the neighborhood (**Figure 4B**). This suggests a well-defined optimum length scale where a clear difference between the two samples occurs, and it is likely to be correlated with relevant dimensions in the sample (e.g., cell size). It is important to note that the optimal length scale will likely exist for a range of different window/neighborhood pairs, which together will define a similar local area. In the example shown in **Figure 4B**, the order parameter difference peaks for a length scale of ~ 90 μm, as calculated by multiplying the window size by the neighborhood size for pairs displaying high difference values. This suggests that this length scale

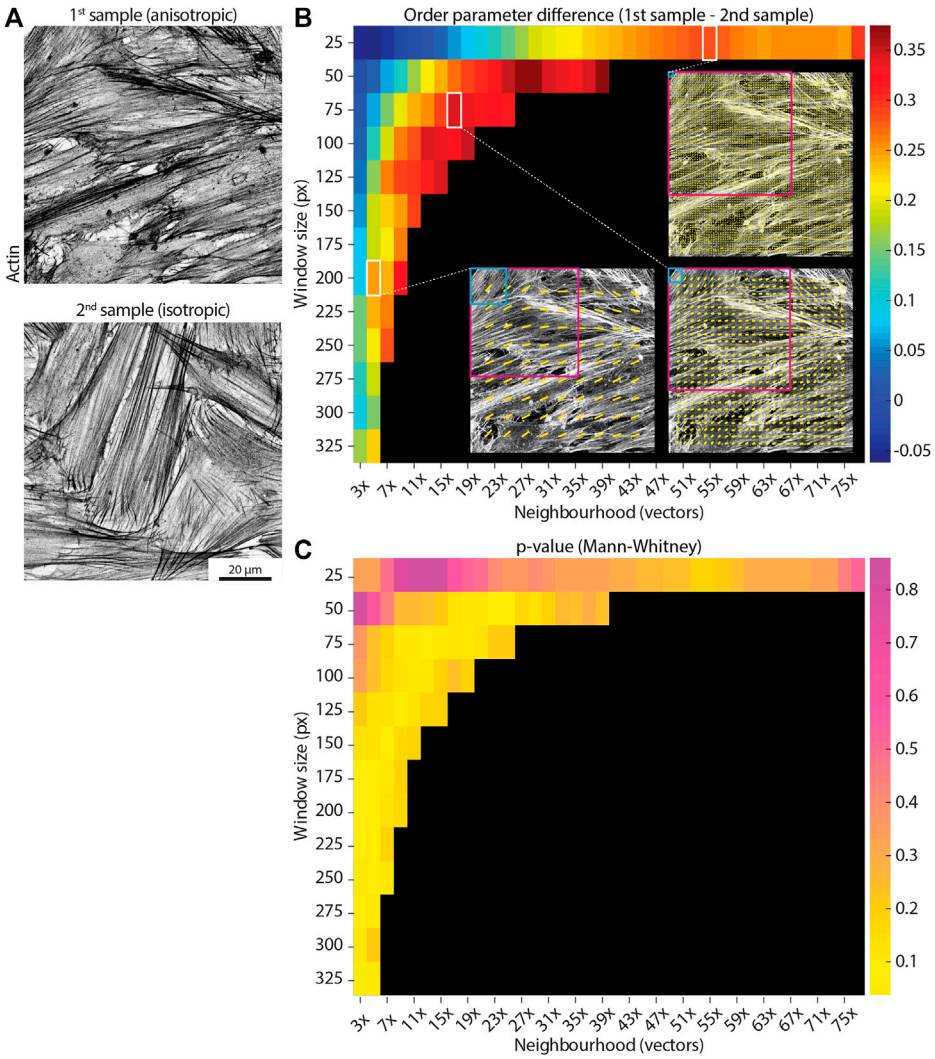


FIGURE 4 | Determining the optimal length scale difference. **(A)** Example actin images taken from two separate samples and displaying different degrees of alignment (higher on the left, more anisotropic; lower on the right, more isotropic). Scale bar, 20 *µm*. **(B)** The parameter search runs the analysis for a range of window sizes (y-axis, from 25 *px* to 325 *px*) and a range of neighborhood sizes (*x*-axis, from 3*x* to 77*x* vectors, where the neighborhood size is defined as 2**neighborhood radius* + 1). Each permutation of window and neighborhood represents a length scale. Each colored square shows the difference in the median order parameter between the anisotropic sample and the isotropic sample (mean values over 10 images for each sample) for a specific pair of window and neighborhood sizes. Small window sizes paired with small neighborhoods (upper left corner) display noisy output, due to the limited amount of image information available to process. Three example vector fields are shown for a window size of 25 *px* and neighborhood of 55*x* vectors (corresponding to 92 *µm*), 75 *px* and 17*x* vectors (89 *µm*), and 200 *px* and 5*x* vectors (79 *µm*), with the light blue and magenta squares showing the size of the window and neighborhood respectively. Despite relatively similar neighborhood sizes, the output varies depending on the size of the examined window. Reading the graph horizontally from left to right for each window size up to ~ 150 *px*, it is possible to note how an optimal length scale of ~ 90 *µm* is detected (i.e., a peak difference value is displayed), likely correlated to relevant biological dimensions (e.g., cell size). **(C)** A similar graph to the one in **(B)** is shown for the *p*-value of Mann-Whitney tests between the two samples for all the combinations of window and neighborhood sizes. In this case, greater differences are represented by lower values (yellow).

should be investigated further to evaluate the biological relevance of features of such size.

3.3 Evaluating the Length Scale Decay in Alignment

A similar approach to the parameter search can be employed to look at how the alignment decays over increasingly larger length scales, by progressively enlarging the size of the region over which the alignment is measured. To this aim, the window size over which the angle vector field is calculated is kept constant, while progressively increasing the neighborhood over which the order parameter is evaluated.

To confirm the performance of the algorithm in locating the alignment length scale, we created three 10-image sets of synthetic data. These images show a number of squared units of pre-defined size (i.e., 100, 200, or 400 *px*), randomly arranged in the field of view. Each unit is filled with parallel lines

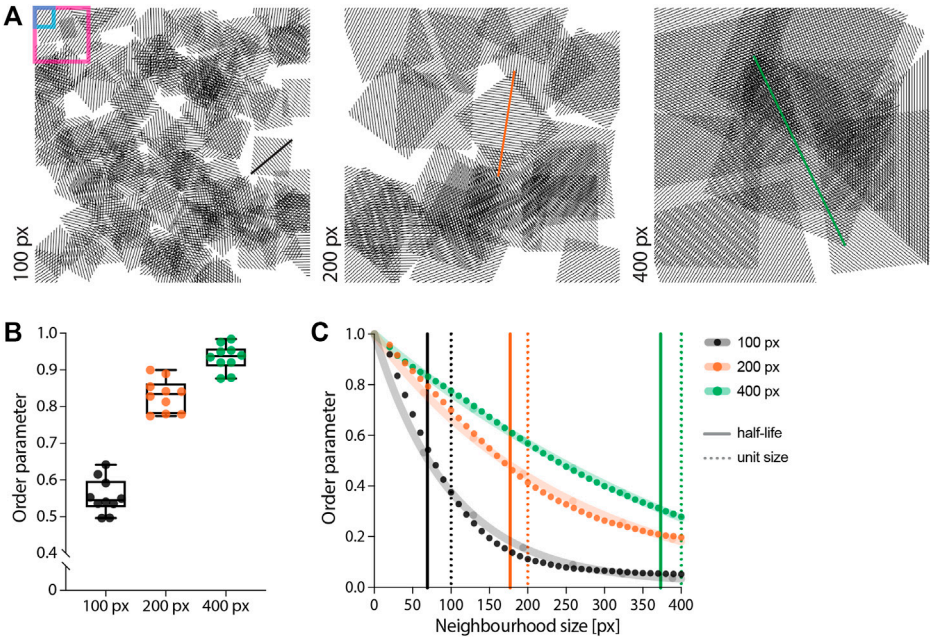


FIGURE 5 | Evaluating length scale on synthetic data. **(A)** Synthetic images were created by randomly distributing units of parallel lines within the field of view. The size of each unit was pre-defined to be either 100 *px* (left, black), 200 *px* (centre, orange) or 400 *px* (right, green). Each sample contains 10 images. The blue and magenta boxes represent the window and neighborhood sizes used for the analysis in **(B)**, set to 35 *px* and 100 *px*, respectively. **(B)** Measuring the order parameter using the same parameters for the three sets of images shows an increase in its median value with the unit size. This can be expected, as the information within this length scale will be more aligned the bigger the unit size. **(C)** Keeping the window size constant (10 *px*) and progressively increasing the neighborhood shows a decay in the order parameter for all three samples (dots). By fitting a one-phase decay to each data set (shaded lines) and measuring its half-life, it is possible to appreciate how the algorithm is able to locate the pre-defined length scale (i.e., unit size).

representing fibers (**Figure 5A**). We first measured the median order parameter for each image in each set with a window size of 35 *px* and a neighborhood of 5*x* vectors, corresponding to 100 *px* (**Figure 5B**). The order parameter increases with the selected unit size, as expected as neighborhoods of a given size will represent a progressively more local point of view for increasing unit sizes. Moreover, we analyzed the decay in the order parameter by keeping a fixed 10 *px* window size and progressively increasing the neighborhood size (**Figure 5C**). The half-life of the fitted decay curve shows values very close to the pre-defined unit size for each set. This result highlights the ability of our approach to correctly identify the alignment length scale within an image.

We then used the alignment decay analysis to quantify the length scale on real biological data, as exemplified by using ~ 1 mm^2 tilescan images of cell-derived matrices (i.e., matrices produced *in situ* by the cells) stained for F-actin and fibronectin (before and after decellularization, respectively, **Figure 6A**). We first calculated the angle vector field for both actin and fibronectin using the FFT approach for a window size of 250 *px* and overlap of 50% (**Figure 6B**). Subsequently, we calculated the median order parameter over neighborhoods of increasing radii, ranging from 1*x* to 21*x* vectors, corresponding to 50 *x* 50 μm to 1050 *x* 1050 μm in the original image (**Figures 6A,C,D**). By plotting the median order parameter over the neighborhood size, it is possible to observe the alignment decaying from the local to the global length scale (**Figure 6D**). As expected, actin and fibronectin displayed a similar decay in alignment as cells are responsible for depositing and remodelling the ECM in this experiment.

3.4 Filtering Input for Alignment Analysis

Some biological images might display regions that are either blank (i.e., little to no signal, lack of image features), or isotropic (i.e., no obvious fibrillar features). In certain applications, such regions should be excluded from the alignment analysis, as images containing noisy or blank areas will affect the output order parameter. By exploiting our window-based algorithm, it is possible to implement optional filtering of features to automatically exclude regions with such characteristics from the analysis.

To filter out blank regions, a threshold can be set on the mean intensity for each window, as demonstrated on second harmonic generation imaging of tissue samples (**Figure 7A**). In this example, windows that are considered background have a mean intensity signal lower than 20 (with intensity values ranging from 0-black to 255-white). This value can be set as a threshold: windows displaying mean intensity values lower than the threshold will be ignored during the calculation of the angle vector field (and subsequent alignment score, **Figure 7B**).

Regions in which image features display isotropic organization can also be excluded by setting a threshold on the eccentricity of the FFT calculated for a given window, as shown for cell derived matrices immunostained for fibronectin in **Figure 7C**. The FFT eccentricity can be used as a measure of its skewness (**Figure 1A**):

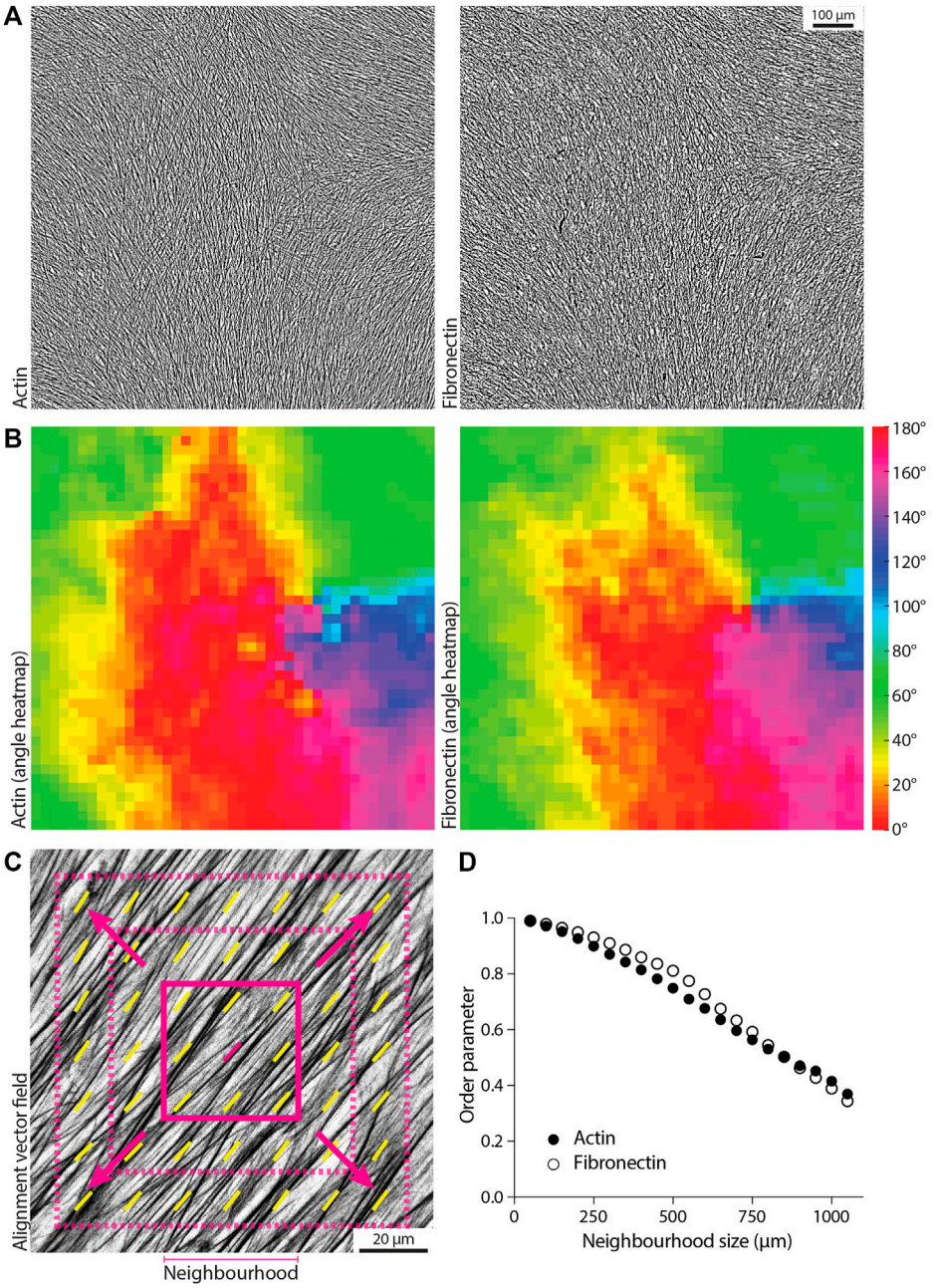


FIGURE 6 | Measuring the decay in order parameter as a function of length scale. **(A)** Example tilescan images about 1 mm^2 in size for actin and fibronectin on the same sample (fibroblast-derived matrix). Scale bar, 100 μm. **(B)** Angle heatmap showing the alignment vector field for the images in **(A)**, using a window size of 250 *px* and 50% overlap. Actin and fibronectin display similar alignment. **(C)** Schematic of the alignment decay analysis, where the window size is maintained constant while increasing the size of the neighborhood (i.e., including more and more vectors in the order parameter calculation to evaluate alignment from the local to the global scale). The solid magenta square represents the starting neighborhood, which then increases in size at each iteration (dashed magenta squares). **(D)** The calculated order parameter for actin and fibronectin for the images in **(A)** is graphed for a window size of 250 *px*, for each neighborhood ranging from 50 μm (3*x* vectors, neighborhood radius of 1*x* vector) to 1050 μm (43*x* vectors, neighborhood radius of 21*x* vectors). Alignment decays in a similar manner when going from local (small neighborhood size) to global (large neighborhood size) length scales.

regions containing oriented fibers will display a more elongated FFT, hence higher eccentricity (values closer to 1); regions where no fibers can be detected, or where there are multiple fibers without a clear orientation, will display a more homogeneously shaped (i.e., round) FFT, hence lower eccentricity (values closer to 0). In the example shown in **Figure 7C**, threshold on the eccentricity was set to 0.73, allowing for the exclusion of isotropic regions from the angle vector field calculation (**Figure 7D**).

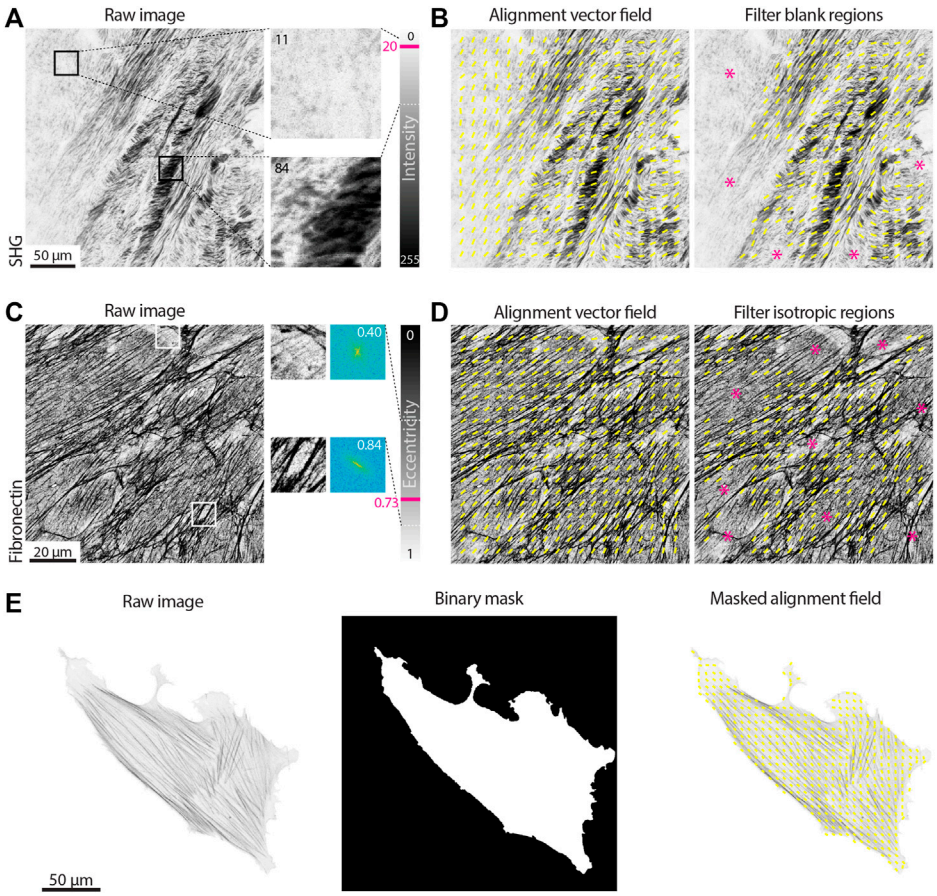


FIGURE 7 | Filtering vector fields based on intensity and eccentricity. **(A)** A tissue section second harmonic generation (SHG) image is shown. Two regions with the same size of the window used for the alignment analysis (100 *px*) are enlarged and their mean intensity value measured. The intensity of an 8-bit image can vary from 0 to 255. The area on the top left contains little to no signal and it is considered background (mean intensity of 11); the area on the right contains many dark pixels, for a mean intensity of 84. A threshold value of 20 is set, meaning that all windows with mean intensity lower than such values will be considered as blank (i.e., background) and excluded from the analysis. Scale bar, 50 *µm*. **(B)** The alignment vector field (yellow) is overlaid on the raw SHG image, either without filtering for blank regions **(left)** or filtering with a threshold on the mean intensity of 20 **(right)**. Magenta asterisks highlight regions excluded from the analysis. **(C)** An immunostaining for fibronectin in cell-derived matrices is shown. Two regions with the same size of the window used for the alignment analysis (100 *px*) are enlarged and their 2D Fast Fourier Transform (FFT) displayed. The FFT eccentricity can range from 0 (circular) to 1 (highly elongated). The region on the top lacks clear aligned fibers, resulting in a more uniform FFT (eccentricity of 0.40); the area on the bottom has aligned fibrillar features, leading to a highly skewed FFT (eccentricity of 0.84). A threshold value of 0.73 is set, meaning that all windows with eccentricity lower than such values will be considered as isotropic and excluded from the analysis. Scale bar, 20 *µm*. **(D)** The alignment vector field (yellow) is overlaid on the raw fibronectin image, either without filtering for isotropic regions **(left)** or filtering with a threshold on the eccentricity of 0.73 **(right)**. Magenta asterisks highlight the regions excluded from the analysis. **(E)** The actin in a single cell is shown. If one wants to exclude the background from the analysis, a binary mask can be provided highlighting the regions of the image to be analyzed (window size 50 *px* with 50% overlap). Scale bar, 50 *µm*.

Finally, it is possible that only a specific region within an image is to be analyzed, as in the case of actin in single cells (**Figure 7E**). A binary image can be used to mask the angle vector field, in order to only consider a given area of the original input.

3.5 Extracting Kinetics of Organization From Live Imaging

Many biological processes involve the evolution of architectures and organizations over time. By measuring an order parameter for each frame of a time series it is possible to extract the kinetics of alignment in response to perturbations. As an example, we transfected an NIH 3T3 fibroblast with GFP-actin and imaged the cell before, during, and after treatment with the Rho kinase inhibitor Y-27632 (**Figure 8**). Inhibition of Rho kinase leads to a reduction in myosin activity and thus a dissolution of stress fibers (e.g., a loss of fibrillar structures; **Figure 8A**). As a test case, treatment with Y-27632 is convenient because it acts rapidly and is easily washed out by replacing the media in the sample chamber with fresh media. Upon removal of Y-27632, the actin cytoskeleton begins to reform stress fibers and again takes on an aligned appearance (**Figure 8B**). As evidenced in the plot of the order parameter, we can see a loss of alignment immediately after addition of Y-27632 and an immediate recovery following washout (**Figure 8C**).

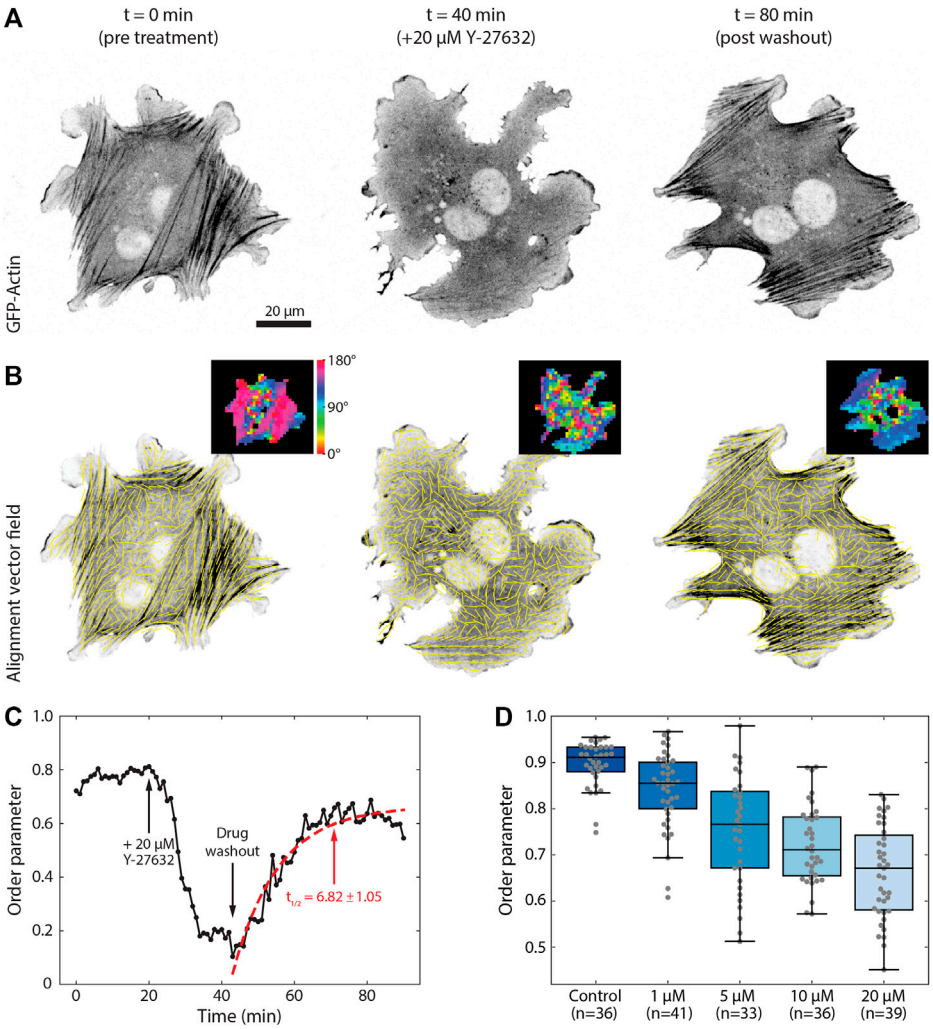


FIGURE 8 | Measuring kinetics of alignment. **(A)** An NIH 3T3 fibroblast transfected with GFP-actin is shown before, during, and after washout of 20 μM Y-27632. **(B)** Local alignment was calculated for each frame in the time series using a window size of 17 *px* and an overlap of 50%. Yellow vectors indicate the local direction of alignment in the image while insets show the angle of orientation for each window. **(C)** Plot of the calculated order parameter over time for a neighborhood radius of 2*x* vectors. Arrows indicate the time points of addition of Y-27632 and wash out of the drug. Red dashed line indicates the fit to **Eq. 7**. **(D)** Boxplots of the average order parameter for fibroblasts treated for 30 min with different concentrations of Y-27632 and then fixed and stained with phalloidin to label actin. Each dot represents the median order parameter of a single cell with a window size of 33 *px*, overlap of 50%, and neighborhood radius of 2*x*.

To measure the kinetics of this interaction we can fit the data to any relevant model. For the washout of Y-27632 we fit the post-washout data to a simple exponential recovery curve of the form

$$S(t) = A - B\exp(-t/C). \tag{7}$$

We can then define a $t_{1/2}$ value, or the time it takes to reach half its maximum recovery value, as

$$t_{1/2} = -C\ln\left(\frac{A}{2B}\right). \tag{8}$$

This results in a value of $t_{1/2} \approx 7$ min for the Y-27632 washout experiment, consistent with previous reports (Aratyn-Schaus et al., 2011). To further illustrate the robustness of this approach, we imaged numerous cells treated with varying concentrations of Y-27632 (**Figure 8D**). Cells were treated with concentrations ranging from 0–20 μM for 30 min, and then fixed and stained with phalloidin. Analysis of these data sets reveals a steady decrease in order parameter as a function of Y-27632 concentration, as expected. The difference in magnitude of the order parameter between the fixed and live imaging is the product of superior signal to noise and better labelling that is achieved with phalloidin compared to fluorescently expressed actin.

3.6 Comparing With Other Available Algorithms

The performance of the algorithm presented in this work was tested against two other open-source tools widely used by the

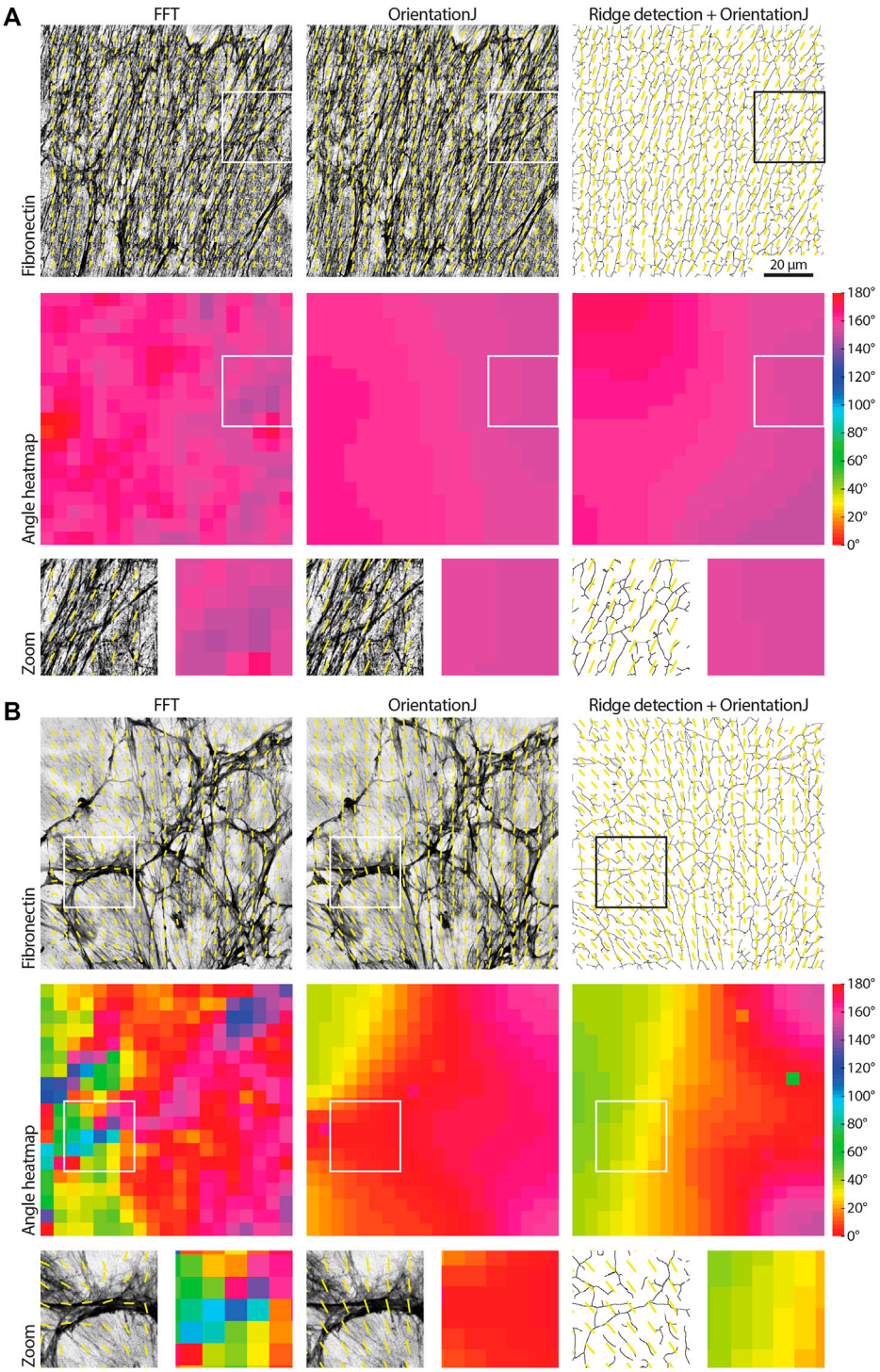


FIGURE 9 | Comparing approaches to measure organization in biological images. **(A)** A fibronectin image showing fibrillar features with strong alignment is analyzed with the FFT approach presented in this work, with the Fiji OrientationJ Vector Field plugin, and with a combined approach using the Fiji Ridge Detection plugin followed by OrientationJ (mimicking the TWOMBLI hybrid approach, the binary mask calculated by the Ridge Detection is shown). A window size of 100 *px* with 50% overlap was used, and the order parameter was calculated over neighborhoods of 5*x* vectors (neighborhood radius of 2*x* vectors). Both the alignment vector field and the angle heatmap are shown for each case; zooms for the areas highlighted by the white boxes are displayed. The three algorithms behave similarly. Scale bar, 20 *µm*. **(B)** The analysis of a more randomly organized fibronectin image is compared as in **(A)**. It can be noted how the FFT returns more accurate alignment vectors for the more isotropic regions of the image (insets).

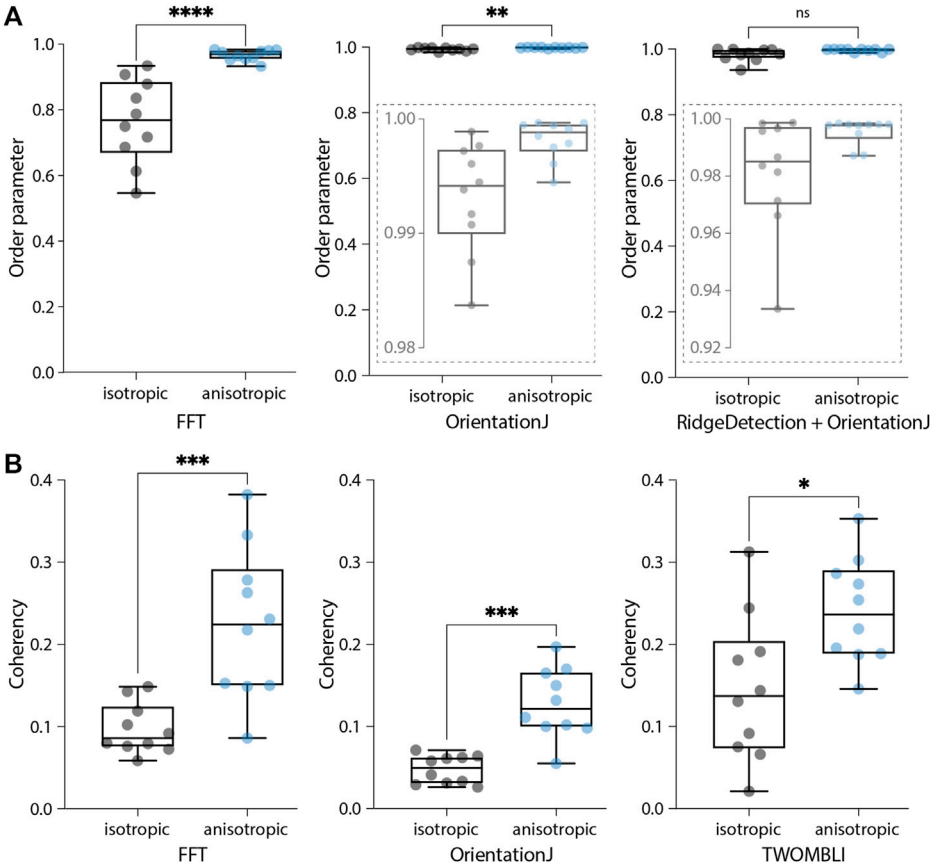


FIGURE 10 | Comparing output metrics of different approaches to measure organization in biological images. **(A)** Boxplots of the order parameters for the samples shown in **Figure 9** calculated with the three methods, the FFT (as presented here), OrientationJ (Vector Field plugin), and the hybrid approach (Ridge Detection followed by OrientationJ Vector Field). Local windows of 100 *px* with 50% overlap and 5*x* vector neighborhoods were considered. Insets represent zooms of the y-axis to better display distributions. Mann-Whitney two-tailed test (****$p < 0.0001$, **$p = 0.0039$, ns = 0.063). **(B)** Boxplots of the global coherency calculated with the three tools, *AFT – Alignment by Fourier Transform* (as presented here), OrientationJ (Dominant Direction plugin), and TWOMBLI. Mann-Whitney two-tailed test (***$p = 0.0002$, ***$p = 0.0002$, *$p = 0.0147$).

biology community, OrientationJ (Rezakhaniha et al., 2012) and TWOMBLI (Wershof et al., 2021). OrientationJ is a Fiji plugin based on image representation by structure tensors using real space, and evaluates orientation by the coherency metric, that indicates the degree of feature orientation with values ranging from 0 (random) to 1 (perfectly aligned) (Rezakhaniha et al., 2012; Clemons et al., 2018). Using this approach, it is possible to output either an alignment vector field (OrientationJ Vector Field) or a global coherency score for the image (OrientationJ Dominant Direction). TWOMBLI is a Fiji macro envisioned as a hybrid approach performing fiber segmentation followed by alignment analysis. Fiber segmentation is carried out with the Ridge Detection plugin (Lindeberg, 1998), while the alignment of the binarized image is subsequently measured *via* OrientationJ. In addition to the coherency metric as a measure of alignment and due to the segmentation step, TWOMBLI also allows the user to obtain further metrics that characterize the fibrillar features, such as number of branch and end points, density, curvature, and fractal dimensions.

We first compared the alignment vector field calculated with the three methods (FFT, structure tensor, and hybrid approach) for cell-derived matrices immunostained for fibronectin displaying different degrees of isotropy (**Figure 9**). When comparing the alignment vector field for images with strongly anisotropic fibrillar features, similar output was observed for the three methods (**Figure 9A**). When working with less aligned input, however, the FFT was able to better resolve the feature alignment in the isotropic regions compared to the other two algorithms (**Figure 9B**).

Two alignment metrics were analyzed, the order parameter *S* (**Eq. 6**) and the coherency; the first metric is the one used in this work, while the coherency is the standard output of both OrientationJ and TWOMBLI. The comparison dataset contained two samples of images, with isotropic (**Figure 9B**) and anisotropic (**Figure 9A**) features. All the tested algorithms were able to recognize the alignment difference in the data, however, the order parameter scores calculated with the TWOMBLI approach (fiber segmentation via Ridge Detection followed by alignment with OrientationJ) did not reach statistical significance (**Figure 10A**). Moreover, the order parameter values calculated with both OrientationJ and the hybrid approach were

TABLE 1 | A comparison of the performance and characteristics of the three analysis methods.

	AFT	OrientationJ	TWOMBLI
Computational time	low (1.8-4.8 s[a])	low (4.8 s[b])	high (72.4 s)
Length scale analysis	[c]	[d]	-
Additional metrics	-	[d]	[c]
Automated image filtering	[c]	-	-

Computational time was measured on a 3.6 GHz Quad-Core Intel Core i7 machine with 32 GB memory for a 10-image sample.
[a]*Depends on window size.*
[b]*With a dedicated macro.*
[c]*Yes*
[d]*Yes, but requires further post-processing.*

very close to the upper limit of the range (values equal to 1 signify perfect alignment). This suggests that these methods might not be reliable in detecting subtle differences between samples, due to their limited ability to resolve local isotropy (**Figure 9**). Global coherency obtained with the three methods was also compared, showing that all of them could successfully distinguish the global alignment differences in the two samples (**Figure 10B**).

Computational time is comparable between the FFT and OrientationJ (with a dedicated macro) with each image being evaluated in less than half a second, while it increases for TWOMBLI with about 7 s per image (due to the fiber segmentation step). While this difference might appear insignificant for small samples as the ones used in this context, it might become important should batch processing be required for larger sample sizes or more high-throughput applications. The FFT approach presented here can be easily used to address length scale analysis. While OrientationJ also offers the option of a length scale analysis (Vector Field plugin), this requires *ad hoc* macros and further post-processing to obtain a unique alignment score for each image for a given window size. TWOMBLI allows the user to obtain a wider range of metrics characterizing the fibrillar features, thanks to its fiber segmentation step, but this comes at the cost of setting more parameters to start the analysis. The FFT performs best at resolving local alignment differences (**Figure 9**), and the available filters on window intensity and eccentricity make it straightforward to automatically exclude regions of the images. This avoids having to manually discard entire images containing unsuitable regions, as would happen with the other two methods. The pros and cons of each methodology should be taken into account depending on the application (**Table 1**); overall, the *AFT* approach demonstrated efficient performance and broad flexibility to different types of input images.

3.7 Implementing a Streamlined Software for Measuring Alignment

Finally, we have developed a workflow named *AFT – Alignment by Fourier Transform* to automate image analysis using the approach described above. This includes the pipeline described in **Section 3.1**, the parameter search in **Section 3.2**, the analysis of length scale decay in **Section 3.3**, the filtering options in **Section 3.4**, and the extraction of organization kinetics in **Section 3.5**. Open-source implementation is made available in both MATLAB and Python at https://github.com/OakesLab/AFT-Alignment_by_Fourier_Transform.

Some considerations regarding the choice of parameters follows. The user is requested to set three mandatory parameters in order to run the alignment analysis: the window size, the overlap, and the neighborhood size (**Section 3.1**, **Figures 1D**, **2B**). Each of these parameters will affect the calculated order parameter by changing the length scale of the alignment analysis. While default parameters are suggested, it is important to test different values on individual images to gain some insight on the optimal length scale over which to carry out this analysis. In addition to the mandatory parameters, a number of optional filtering features are available as described in **Section 3.4** (**Figure 7**).

3.7.1 Mandatory Parameters

- *Window size.* The size of the local regions in the image to analyse. This value depends on the image resolution, the size of the fibrillar features to be examined, and the length scale of interest (**Figure 1D**). In practice, within a window one should be able to observe enough information to discern by eye an orientation of features. There is no hard-set lowest limit for the window size, but it should be noted that smaller windows will lead to more spurious vectors as the calculated FFT will be nearly circular (i.e., not skewed, **Figure 1A**) for regions where fibrillar features are not recognized. A visual check of the output images is always recommended during the parameter optimization phase.
- *Overlap.* The overlap between adjacent windows (**Figure 1D**). Increasing the overlap, in conjunction with choosing smaller window sizes, allows for increasing the resolution of the analysis, with more sampled areas (i.e., vectors) within the field of view. A starting value of 50% is recommended.
- *Neighborhood radius.* The length scale over which to compare neighboring measures of alignment. This should be set depending on the length scale of interest. It is defined as the number of nearest neighbor vectors to be compared to a central reference in order to obtain an alignment score (**Figure 2**). Neighborhood size is calculated as 2**neighborhood radius* + 1.

3.7.2 Optional Parameters

- *Output images.* This parameter enables the ability to save the analyzed images, consisting of the original image overlaid with the alignment vector field and the angle heatmap (**Figure 1D**). The color scale for the angle heatmap is wrapped, with similar colors for 0° and 180°, as fibers are not considered polar (**Section 3.1**, **Figure 1D**).
- *Filter blank spaces.* Windows that contain little or no signal can be excluded from the analysis. When enabled a threshold on the mean window intensity can be set (**Figures 7A,B**). Windows below this mean intensity are not analyzed. Threshold values can be estimated by opening a sample image in Fiji, selecting regions considered

background with similar size to the window size, and measuring their mean intensity. Expected values range from 0 (black) to 255 (white) for an 8-bit image.

- *Filter isotropic regions*. It is possible to exclude windows from the analysis should the eccentricity of their FFT be lower than a threshold (**Figures 7C,D**). Values for the FFT eccentricity range from 0 (circular, isotropic) to 1 (elongated, anisotropic). To estimate this threshold, the analysis can be run iteratively for increasing values until the desired output is achieved (i.e., until the regions considered isotropic do not contain alignment vectors in the output images).
- *Masking*. By default, the whole image is being analyzed. Alternatively, the user can input a folder containing binary masks of selected areas to be analyzed for each input image (**Figure 7E**). A similar output as the one shown in **Figure 7E** could have also been obtained by filtering out the blank spaces, as the regions of interest have brighter signal than the background. However, if regions devoid of signal would also be present inside the cell outline, they would be disregarded as well. Therefore, a binary masking approach might be preferable in this case.

4 DISCUSSION

Quantifying alignment of fibrillar features in biological images has gained wide interest as a possible biomarker in disease aetiology and progression (Ouellette et al., 2021). While tools are available for this aim (Püspöki et al., 2016; Liu et al., 2017), many are difficult to implement and fairly rigid on the length scale over which anisotropy can be interrogated. The method presented in this work exploits the representation of the image in the frequency domain paired with a custom window-based approach, offering a computationally efficient algorithm that can be easily applied to investigate local to global alignment. Decay in alignment scores with increasing distance can be evaluated to reveal the length scale at which fibrillar features remain anisotropic.

An open-source suite, *AFT – Alignment by Fourier Transform*, is presented: its flexibility to a wide range of biological images is showcased through the examples, and its performance tested against synthetic data. The *AFT – Alignment by Fourier Transform* tool is demonstrated to be robust for different imaging modes (fixed, live-cell, SHG, fluorescence), fluorescent probes (F-actin, fibronectin), and image resolutions (from single cells to tissue tilescans). Further applications for which this method could be used that were not shown in the present work could entail the evaluation of alignment across the depth of a sample (measuring alignment for each slice of a Z-stack), or the investigation of cytoskeletal architectures in reconstituted protein systems.

The FFT approach performance was comparable to both OrientationJ and TWOMBLI, with increased accuracy in resolving local isotropic regions. Its computational efficiency makes it a good candidate for analyzing high volumes of data. Moreover, the availability of filters and masking options allow for this methodology to be applied to a wide range of microscopy images.

Excitingly, more advanced 3-dimensional imaging modalities are becoming available, with techniques being developed to achieve isotropic resolution. While currently the FFT approach is designed to work with 2D data, the algorithm could be adapted to evaluate alignment of 3D fibrillar features, and its low computational cost could be exploited for these more computationally intensive data.

5 CONCLUSION

Here we presented a methodology to measure fibrillar feature alignment in biological images. We created an open-source tool that can be used on a wide range of biological images. Its performance was compared against other common workflows, and it was shown to be computationally efficient without compromising on accuracy. It is easy to implement with a small number of analysis parameters, and allows for interrogating the length scale of fiber anisotropy.

DATA AVAILABILITY STATEMENT

All the experimental datasets used as examples are included in the Supplementary Material (https://doi.org/10.6084/m9.figshare.15326472.v1), further inquiries can be directed to the corresponding authors. The software implementations can be found at https://github.com/OakesLab/AFT-Alignment_by_Fourier_Transform, together with further documentation on how to run the code.

AUTHOR CONTRIBUTIONS

Conceptualization and Methodology, PO; Software, SM, DF, PO; Data acquisition, LT, FK; Formal analysis and Data curation, SM, PO; Writing SM, DF, LT, FK, TS, BS, PO; Funding Acquisition and Supervision, TS, BS, PO.

FUNDING

This project has been funded from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 681808 - SM, BS), the Wellcome Trust (grant no. 107859/Z/15/Z - FK, BS), the London Interdisciplinary Doctoral Programme (BB/J014567/1 - DBDF), the National Institutes of Health (NIH) National Institute of Allergy and Infectious Disease (NIAID) (Award # P01 AI02851 - LDT, PWO), and the National Science Foundation (NSF) (CAREER award # 2000554 - PWO). For the purpose of open access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://doi.org/10.6084/m9.figshare.15326472.v1

REFERENCES

Aratyn-Schaus, Y., Oakes, P. W., and Gardel, M. L. (2011). Dynamic and Structural Signatures of Lamellar Actomyosin Force Generation. *MBoC* 22, 1330–1339. doi:10.1091/mbc.e10-11-0891

Boudaoud, A., Burian, A., Borowska-Wykręt, D., Uyttewaal, M., Wrzalik, R., Kwiatkowska, D., et al. (2014). FibrilTool, an ImageJ Plug-In to Quantify Fibrillar Structures in Raw Microscopy Images. *Nat. Protoc.* 9, 457–463. doi:10.1038/nprot.2014.024

Bredfeldt, J. S., Liu, Y., Conklin, M. W., Keely, P. J., Mackie, T. R., and Eliceiri, K. W. (2014a). Automated Quantification of Aligned Collagen for Human Breast Carcinoma Prognosis. *J. Pathol. Inform.* 5, 28. doi:10.4103/2153-3539.139707

Bredfeldt, J. S., Liu, Y., Pehlke, C. A., Conklin, M. W., Szulczewski, J. M., Inman, D. R., et al. (2014b). Computational Segmentation of Collagen Fibers from Second-Harmonic Generation Images of Breast Cancer. *J. Biomed. Opt.* 19, 016007. doi:10.1117/1.jbo.19.1.016007

Cetera, M., Ramirez-San Juan, G. R., Oakes, P. W., Lewellyn, L., Fairchild, M. J., Tanentzapf, G., et al. (2014). Epithelial Rotation Promotes the Global Alignment of Contractile Actin Bundles during Drosophila Egg Chamber Elongation. *Nat. Commun.* 5, 1. doi:10.1038/ncomms6511

Chaudhuri, S., Nguyen, H., Rangayyan, R. M., Walsh, S., and Frank, C. B. (1987). A Fourier Domain Directional Filterng Method for Analysis of Collagen Alignment in Ligaments. *IEEE Trans. Biomed. Eng.* BME-34, 509–518. doi:10.1109/TBME.1987.325980

Clemons, T. D., Bradshaw, M., Toshniwal, P., Chaudhari, N., Stevenson, A. W., Lynch, J., et al. (2018). Coherency Image Analysis to Quantify Collagen Architecture: Implications in Scar Assessment. *RSC Adv.* 8, 9661–9669. doi:10.1039/c7ra12693j

Duclos, G., Erlenkämper, C., Joanny, J.-F., and Silberzan, P. (2017). Topological Defects in Confined Populations of Spindle-Shaped Cells. *Nat. Phys* 13, 58–62. doi:10.1038/nphys3876

Duclos, G., Garcia, S., Yevick, H. G., and Silberzan, P. (2014). Perfect Nematic Order in Confined Monolayers of Spindle-Shaped Cells. *Soft Matter* 10, 2346–2353. doi:10.1039/c3sm52323c

Eltzner, B., Wollnik, C., Gottschlich, C., Huckemann, S., and Rehfeldt, F. (2015). The Filament Sensor for Near Real-Time Detection of Cytoskeletal Fiber Structures. *PLoS ONE* 10, e0126346–28. doi:10.1371/journal.pone.0126346

Falzone, T. T., Oakes, P. W., Sees, J., Kovar, D. R., and Gardel, M. L. (2013). Actin Assembly Factors Regulate the Gelation Kinetics and Architecture of F-Actin Networks. *Biophysical J.* 104, 1709–1719. doi:10.1016/j.bpj.2013.01.017

Fernandes, N. R. J., Reilly, N. S., Schrock, D. C., Hocking, D. C., Oakes, P. W., and Fowell, D. J. (2020). CD4+ T Cell Interstitial Migration Controlled by Fibronectin in the Inflamed Skin. *Front. Immunol.* 11, 1501–1514. doi:10.3389/fimmu.2020.01501

Goldyn, A. M., Kaiser, P., Spatz, J. P., Ballestrem, C., and Kemkemer, R. (2010). The Kinetics of Force-Induced Cell Reorganization Depend on Microtubules and Actin. *Cytoskeleton* 67, NA. doi:10.1002/cm.20439

Gupta, M., Doss, B. L., Kocgozlu, L., Pan, M., Mège, R.-M., Callan-Jones, A., et al. (2019). Cell Shape and Substrate Stiffness Drive Actin-Based Cell Polarity. *Phys. Rev. E* 99, 1–16. doi:10.1103/PhysRevE.99.012412

Gupta, M., Sarangi, B. R., Deschamps, J., Nematbakhsh, Y., Callan-Jones, A., Margadant, F., et al. (2015). Adaptive Rheology and Ordering of Cell Cytoskeleton Govern Matrix Rigidity Sensing. *Nat. Commun.* 6, 1. doi:10.1038/ncomms8525

Kartasalo, K., Pölönen, R. P., Ojala, M., Rasku, J., Lekkala, J., Aalto-Setälä, K., et al. (2015). CytoSpectre: A Tool for Spectral Analysis of Oriented Structures on Cellular and Subcellular Levels. *BMC Bioinformatics* 16, 1–23. doi:10.1186/s12859-015-0782-y

Lindeberg, T. (1998). Edge Detection and Ridge Detection with Automatic Scale Selection. *Int. J. Comp. Vis.* 30, 117–156. doi:10.1023/a:1008097225773

Linsmeier, I., Banerjee, S., Oakes, P. W., Jung, W., Kim, T., and Murrell, M. P. (2016). Disordered Actomyosin Networks Are Sufficient to Produce Cooperative and Telescopic Contractility. *Nat. Commun.* 7, 12615. doi:10.1038/ncomms12615

Liu, Y., Keikhosravi, A., Mehta, G. S., Drifka, C. R., and Kevin, W. (2017). Methods for Quantifying Fibrillar Collagen Alignment. *Methods Mol. Biol.* 1627, 429–451. doi:10.1007/978-1-4939-7113-8_28

Liu, Y., Keikhosravi, A., Pehlke, C. A., Bredfeldt, J. A., Dutson, M., Liu, H., et al. (2020). Fibrillar Collagen Quantification with Curvelet Transform Based Computational Methods. *Front. Bioeng. Biotechnol.* 8, 1–14. doi:10.3389/fbioe.2020.00198

Livne, A., Bouchbinder, E., and Geiger, B. (2014). Cell Reorientation under Cyclic Stretching. *Nat. Commun.* 5, 1. doi:10.1038/ncomms4938

Marquez, J. P. (2006). Fourier Analysis and Automated Measurement of Cell and Fiber Angular Orientation Distributions. *Int. J. Sol. Structures* 43, 6413–6423. doi:10.1016/j.ijsolstr.2005.11.003

Mascharak, S., des Jardins-Park, H. E., Davitt, M. F., Griffin, M., Borrelli, M. R., Moore, A. L., et al. (2021). Preventing Engrailed-1 Activation in Fibroblasts Yields Wound Regeneration without Scarring. *Science* 372, 1. doi:10.1126/science.aba2374

Moisan, L. (2011). Periodic Plus Smooth Image Decomposition. *J. Math. Imaging Vis.* 39, 161–179. doi:10.1007/s10851-010-0227-1

Ouellette, J. N., Drifka, C. R., Pointer, K. B., Liu, Y., Lieberthal, T. J., Kao, W. J., et al. (2021). Navigating the Collagen Jungle: The Biomedical Potential of Fiber Organization in Cancer. *Bioengineering* 8, 1–19. doi:10.3390/bioengineering8020017

Park, D., Wershof, E., Boeing, S., Labernadie, A., Jenkins, R. P., George, S., et al. (2020). Extracellular Matrix Anisotropy Is Determined by TFAP2C-dependent Regulation of Cell Collisions. *Nat. Mater.* 19, 227–238. doi:10.1038/s41563-019-0504-3

Pourdeyhimi, B., Dent, R., and Davis, H. (1997). Measuring Fiber Orientation in Nonwovens Part III: Fourier Transform. *Textile Res. J.* 67, 143–151. doi:10.1177/004051759706700211

Püspöki, Z., Storath, M., Sage, D., and Unser, M. (2016). "Transforms and Operators for Directional Bioimage Analysis: A Survey," in Focus On Bio-Image Informatics. *Advances in Anatomy Embryology and Cell Biology*. Editors W. De Vos, S. Munck, and J. Timmermans (Cham: Springer), 69–93. doi:10.1007/978-3-319-28549-8_3

Rezakhaniha, R., Agianniotis, A., Schrauwen, J. T., Griffa, A., Sage, D., Bouten, C. V. C., et al. (2012). Experimental Investigation of Collagen Waviness and Orientation in the Arterial Adventitia Using Confocal Laser Scanning Microscopy. *Biomech. Model. Mechanobiology* 11, 461–473. doi:10.1007/s10237-011-0325-z

Ridley, A. J., and Hall, A. (1992). The Small Gtp-Binding Protein Rho Regulates the Assembly of Focal Adhesions and Actin Stress Fibers in Response to Growth Factors. *Cell* 70, 389–399. doi:10.1016/0092-8674(92)90163-7

Sander, E. A., and Barocas, V. H. (2009). Comparison of 2D Fiber Network Orientation Measurement Methods. *J. Biomed. Mater. Res. - A* 88, 322–331. doi:10.1002/jbm.a.31847

Standley, P. R., Camaratta, A., Nolan, B. P., Purgason, C. T., and Stanley, M. A. (2002). Cyclic Stretch Induces Vascular Smooth Muscle Cell Alignment via NO Signaling. *Am. J. Physiol. - Heart Circulatory Physiol.* 283, 1907–1914. doi:10.1152/ajpheart.01043.2001

Wershof, E., Park, D., Barry, D. J., Jenkins, R. P., Rullan, A., Wilkins, A., et al. (2021). A FIJI Macro for Quantifying Pattern in Extracellular Matrix. *Life Sci. Alliance* 4, e202000880. doi:10.26508/lsa.202000880

Willert, C. E., and Gharib, M. (1991). Digital Particle Image Velocimetry. *Experiments in Fluids* 10, 181–193. doi:10.1007/BF00190388

Yoshigi, M., Hoffman, L. M., Jensen, C. C., Yost, H. J., and Beckerle, M. C. (2005). Mechanical Force Mobilizes Zyxin from Focal Adhesions to Actin Filaments and Regulates Cytoskeletal Reinforcement. *J. Cell Biol.* 171, 209–215. doi:10.1083/jcb.200505018

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 Computer Science

ORIGINAL RESEARCH
published: 20 October 2021
doi: 10.3389/fcomp.2021.734559

Automated Deep Lineage Tree Analysis Using a Bayesian Single Cell Tracking Approach

Kristina Ulicna[1,2], *Giulia Vallardi*[1,2], *Guillaume Charras*[2,3,4] *and Alan R. Lowe*[1,2,4,5]*

[1]Institute of Structural and Molecular Biology, University College London, London, United Kingdom, [2]London Centre for Nanotechnology, University College London, London, United Kingdom, [3]Department of Cell and Developmental Biology, University College London, London, United Kingdom, [4]Institute for the Physics of Living Systems, University College London, London, United Kingdom, [5]The Alan Turing Institute, London, United Kingdom

Single-cell methods are beginning to reveal the intrinsic heterogeneity in cell populations, arising from the interplay of deterministic and stochastic processes. However, it remains challenging to quantify single-cell behaviour from time-lapse microscopy data, owing to the difficulty of extracting reliable cell trajectories and lineage information over long time-scales and across several generations. Therefore, we developed a hybrid deep learning and Bayesian cell tracking approach to reconstruct lineage trees from live-cell microscopy data. We implemented a residual U-Net model coupled with a classification CNN to allow accurate instance segmentation of the cell nuclei. To track the cells over time and through cell divisions, we developed a Bayesian cell tracking methodology that uses input features from the images to enable the retrieval of multi-generational lineage information from a corpus of thousands of hours of live-cell imaging data. Using our approach, we extracted 20,000 + fully annotated single-cell trajectories from over 3,500 h of video footage, organised into multi-generational lineage trees spanning up to eight generations and fourth cousin distances. Benchmarking tests, including lineage tree reconstruction assessments, demonstrate that our approach yields high-fidelity results with our data, with minimal requirement for manual curation. To demonstrate the robustness of our minimally supervised cell tracking methodology, we retrieve cell cycle durations and their extended inter- and intra-generational family relationships in 5,000 + fully annotated cell lineages. We observe vanishing cycle duration correlations across ancestral relatives, yet reveal correlated cyclings between cells sharing the same generation in extended lineages. These findings expand the depth and breadth of investigated cell lineage relationships in approximately two orders of magnitude more data than in previous studies of cell cycle heritability, which were reliant on semi-manual lineage data analysis.

Keywords: nuclei segmentation, cell classification, multi-object tracking, lineage tree reconstruction, single-cell heterogeneity

OPEN ACCESS

Edited by:
Virginie Uhlmann,
European Bioinformatics Institute (EMBL-EBI), United Kingdom

Reviewed by:
Steffen Wolf,
MRC Laboratory of Molecular Biology (LMB), United Kingdom
Nico Scherf,
Dresden University of Technology, Germany

***Correspondence:**
Alan R. Lowe
a.lowe@ucl.ac.uk

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *01 July 2021*
Accepted: *22 September 2021*
Published: *20 October 2021*

Citation:
Ulicna K, Vallardi G, Charras G and Lowe AR (2021) Automated Deep Lineage Tree Analysis Using a Bayesian Single Cell Tracking Approach. Front. Comput. Sci. 3:734559. doi: 10.3389/fcomp.2021.734559

1 INTRODUCTION

Individual cells grown in identical conditions within populations of either clonal or closely related origin often exhibit highly heterogeneous proliferative behaviour (Skylaki et al., 2016). Deciphering why and how cell heterogeneity is established, maintained and propagated over generations remains a key challenge (**Figure 1A**). This is increasingly important in studying dynamic developmental processes involving the emergence of diverse cell types committed to different cell fates (Bendall et al., 2014), as well as in pathological scenarios, including cancer (Dagogo-jack and Shaw, 2017).

The contribution of stochasticity and determinism to the origins of cell cycle duration heterogeneity in cultured populations has been examined previously (Sandler et al., 2015; Chakrabarti et al., 2018; Kuchen et al., 2020). However, these analyses have been performed by manually annotating movies, which is laborious and limits the depth and statistical power to study more distant relationships amid noisy data. Despite major efforts in the area of automated cell detection and tracking (Bao et al., 2006; Jaqaman et al., 2008; Downey et al., 2011; Amat et al., 2014; Magnusson et al., 2015; Schiegg et al., 2015; Faure et al., 2016; Hilsenbeck et al., 2016; Skylaki et al., 2016; Stegmaier et al., 2016; Akram et al., 2017; Tinevez et al., 2017; Ulman et al., 2017; Allan et al., 2018; Hernandez et al., 2018; McQuin et al., 2018; Schmidt et al., 2018; Wen et al., 2018; Wolff et al., 2018; Berg et al., 2019; Han et al., 2019; Moen et al., 2019; Tsai et al., 2019; Fazeli et al., 2020; Lugagne et al., 2020; Stringer et al., 2020; Bannon et al., 2021; Fazeli et al., 2021; Mandal and Uhlmann, 2021; Sugawara et al., 2021; Tinevez, 2021), high-fidelity extraction of multi-generational lineages remains a major bottleneck and rate-limiting step in microscopy image analysis. The requirement for additional human oversight to manually curate, or correct, the tracker outputs represents a laborious, time-consuming and often error-prone task. This results in trade-offs being made between the minimum experimental replicates sufficient for reliable low-throughput analysis and maximum volumes of imaging data that researchers are capable to semi-manually process. Further, no single tracking algorithm is likely to be universally performant across all experimental datasets, necessitating an ecosystem of algorithms for scientists to choose from.

To increase the throughput of single-cell studies focusing on cell relationships within their lineages, we developed a hybrid deep learning and cell tracking approach to automatically reconstruct lineage trees from a corpus of live-cell data with single-cell resolution (**Figure 1B**). Our workflow consists of a cell detection step, where individual cells are segmented from live-cell

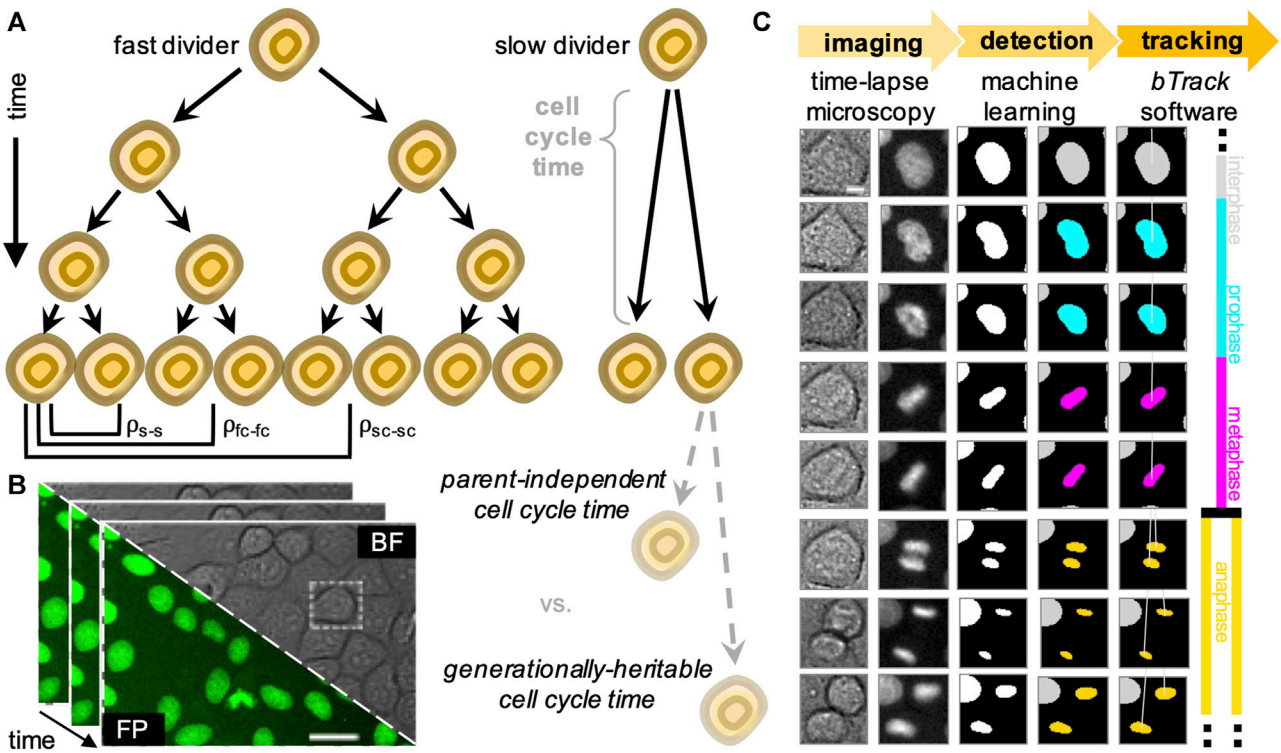


FIGURE 1 | Overview of the experimental (data acquisition) and computational (data analysis) design to track single cells and generate lineage trees. **(A)** We analyse heritability of cell cycle duration across multiple cell generations using automatically reconstructed multi-generational lineage trees. **(B)** Sequential fields of view obtained by live-cell imaging experiments, showing both brightfield (BF) and fluorescence (FP). Scale bar = 20 μ m. **(C)** Fully automated, deep learning-based movie analysis consists of a cell detection step using information from bright-field (col. 1) and fluorescence (col. 2) time-lapse microscopy channels. Cells are localised using the segmentation network (col. 3) and labelled according to their mitotic state by a trained classifier (col. 4). Our software then performs multi-object tracking to reconstruct individual cell trajectories (col. 5) and assembles the parent-children relationships (col. 6) into lineage tree representations. Scale bar = 5 μ m.

images with a wide range of cell densities and fluorescence intensities. The segmented nuclei are subsequently classified according to their cell cycle stage based on their chromatin morphology, followed by a Bayesian tracking algorithm for unsupervised single-cell tracking of cell populations imaged using time-lapse microscopy (**Figure 1C**).

Benchmarking results confirm that our open-source Python pipeline connects single-cell observations into biologically relevant trajectories and correctly identifies cell divisions and relationships within cell families with high fidelity. Enabled by our fully automated approach, we extracted 20,074 single cells organised into 5,325 multi-generational cell lineages with annotated ancestor-descendant relationships in graphical lineage tree representations. This repository of fully annotated cell tracks corresponds to two orders of magnitude more single-cell data than in previous studies of cell cycle heritability (Sandler et al., 2015; Chakrabarti et al., 2018; Kuchen et al., 2020). To demonstrate the utility of our pipeline, we analyse cell cycle durations of our heterogeneous cell population on single-cell level and determine their cross-generational correlations in extended cell lineages.

2 METHODS AND MATERIALS

2.1 Image Acquisition

2.1.1 Automated Widefield Microscopy

A custom-built automated epifluorescence microscope was built inside a standard CO_2 incubator (Heraeus BL20) which maintained the environment at 37°C and 5% CO_2. The microscope utilised an 20× air objective (Olympus Plan Fluorite, 0.5 NA, 2.1 mm WD), high performance encoded motorized XY and focus motor stages (Prior H117E2IX, FB203E and ProScan III controller) and a 9.1 MP CCD camera (Point Grey GS3-U3-91S6M). Brightfield illumination was provided by a fibre-coupled green LED (Thorlabs, 530 nm). GFP and mCherry/RFP fluorescence excitation was provided by a LED light engine (Bluebox Optics niji). Cameras and light sources were synchronised using TTL pulses from an external D/A converter (Data Translation DT9834). Sample humidity was maintained using a custom-built chamber humidifier. The microscope was controlled by MicroManager (Edelstein et al., 2014) and the custom-written software OctopusLite, available at: https://github.com/quantumjot/OctopusLite.

2.1.2 Cell Culture Conditions

We used Madin-Darby Canine Kidney (MDCK) epithelial cells as a model system. Wild-type MDCK cells were grown, plated and imaged as described previously (Norman et al., 2012; Bove et al., 2017). To enable visualisation and tracking of the cells, we used a previously establish MDCK line stably expressing H2B-GFP (Bove et al., 2017). Cells were seeded at initial density of ≈3 × 10^4 cells/cm^2 in 24-well glass-bottom plates (ibidi). The imaging was started 2–3 h after seeding. Imaging medium used during the assay was phenol red free DMEM (Thermo Fisher Scientific, 31053), supplemented with fetal calf serum (Thermo Fisher Scientific, 10270106) and antibiotics.

2.1.3 Time-Lapse Movie Acquisition

We acquired a dataset of 44 long duration (~ 80 h) time-lapse movies of MDCK wild-type cells in culture across nine biological replicates. A typical experiment captured multiple locations for over 80 h with a constant frame acquisition frequency of 4 min for each position. Multi-location imaging was performed inside the incubator-scope for durations of ≈1,200 images (80 h). In total, the dataset comprised 52,896 individual 1,600 × 1,200 × 2 (530 × 400 μm) channel images (brightfield and the nuclear marker H2B-GFP), containing ~250,000 unique cells. This dataset of >3,500 h spans densities from isolated cells to highly confluent monolayers.

2.2 Software Implementation

2.2.1 Image Processing

All image processing was performed in Python, using scikit-image, scikit-learn, Tensorflow 2.4, on a rack server running Ubuntu 18.04LTS with 256 Gb RAM and NVIDIA GTX1080Ti GPUs. The *btrack* package was implemented in Python 3.7 and C/C++ using CVXOPT, GLPK, Numpy and Scipy libraries. We tested the software on OS X, Ubuntu and Windows 10 and found the *btrack* algorithm to be performant even on basic commodity hardware such as laptops. We have provided extensive documentation online, visualisation tools, and user-friendly tutorials with tracking examples, example data and installation instructions.

2.2.2 Data Visualisation Tools

In addition, we developed a track visualization layer and interactive lineage tree plugin for the open-source multi-dimensional image viewer, *napari-arboretum* (Sofroniew et al., 2021) available at: https://github.com/quantumjot/Arboretum. To visualize single cell tracking data using the *napari*-integrated `Tracks` layer, follow the fundamentals tutorial available at: https://napari.org/tutorials/applications/cell_tracking.

2.3 Cell Tracking Pipeline

In this subsection, we provide a detailed description of our step-wise pipeline to process the raw microscopy image data to extract reconstructed lineage tree information in a fully automated manner. However, it should be noted that the steps in our workflow function independently, allowing users to integrate different software tools into our modular pipeline based on their preference. This is including, but not limited to, the popular segmentation algorithms such as CellPose (Stringer et al., 2020) or StarDist (Schmidt et al., 2018).

2.3.1 Cell Instance Segmentation

To localise cells in the raw fluorescence images, we constructed a U-Net (Ronneberger et al., 2015) to first segment cell nuclei in the time-lapse microscopy images (**Figure 2**) using a semantic segmentation approach. We used residual blocks (He et al., 2015) in each of the five convolutional up and down layers (3 × 3 kernels), and uses max-pooling and nearest neighbor upscaling to down and upsample respectively. The final layer of the network is a 1 × 1 convolution with two kernels in the case of a

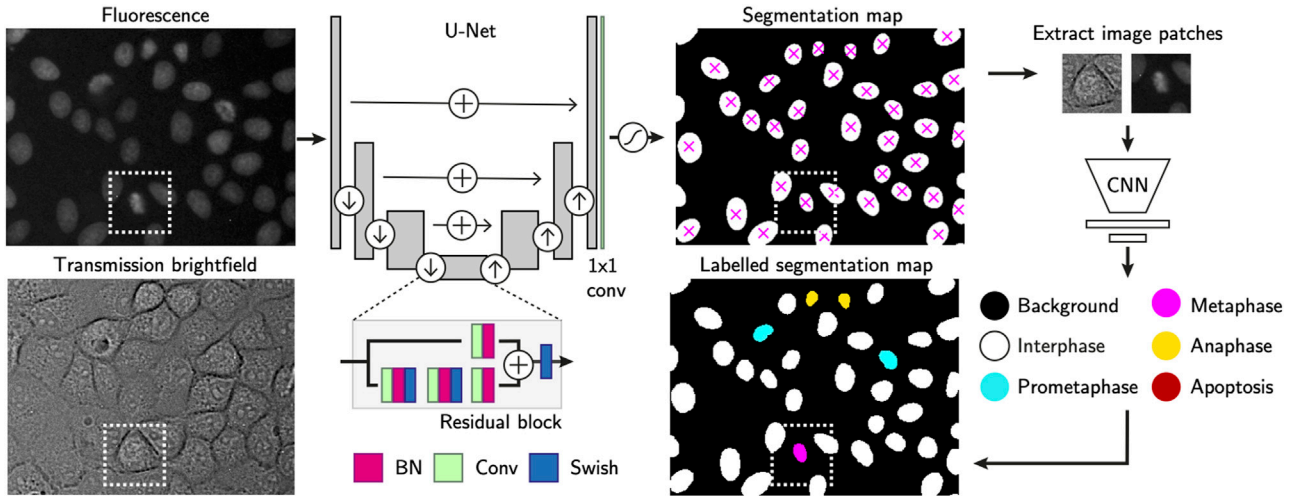


FIGURE 2 | Single-cell annotation workflow. Cell instances are segmented from fluorescence images of their H2B-GFP labelled nuclei. The U-Net consists of residual blocks and residual skip connections. A final 1 × 1 convolution layer, followed by softmax activation generates the output. Detected cells are localised using the centre of mass and used to crop nucleus-centred image patches from both transmission and fluorescence images. These serve as inputs to a CNN-based cell state classifier to label instantaneous phase of each segmented object. Labels indicate whether the cell is in interphase, mitosis or apoptosis. Scale bars = 20 μm.

binary segmentation, with a per-pixel softmax activation. The final segmentation map is decided by arg max function on the stack of class-corresponding maps, outputting the class with highest probability at each position in the FoV. We trained the network with 150 hand-segmented training images with ranging levels of cell confluency, from which regions of 768 × 768 pixels were randomly cropped using on-the-fly augmentations such as cropping, rotation, flipping, noise addition, uneven illumination simulation, scale and affine deformations. At train time, we used a weighted per-pixel cross entropy loss function (Ronneberger et al., 2015) to force the network to prioritise regions separating proximal nuclei. To achieve accurate learning of the border pixels, we constructed pixel-wise weight maps using a difference of Gaussian (DoG) filter on a Euclidean distance map to upweight the importance of pixels at either foreground-background interface or pixels separating two or more proximal cells. We trained the network using the Adam optimizer (Kingma and Ba, 2015) with batch normalisation, a batch size of 16 for 500 epochs after an extensive search for the most optimal hyperparameters. In the following step, each segmentation mask has all of its non-zero values labelled as individual cell instances while zero values are considered the background. This turns the semantic map into instance segmentation mask, in which each cell can be counted and localized by the centre of mass of each segmented region (**Figure 2**).

2.3.2 Cell State Classification

In the following, optional step, the segmentation masks have each single cell region cropped as image patches of 64 × 64 pixels. Those are extracted from the corresponding cell positions in both transmission and fluorescence channel images. Our proprietary cell state classifier (**Figure 2**) is used to label the phase of the current state each cell is in [$s \in$ (Interphase, Prometaphase, Metaphase, Anaphase, Apoptosis)].

The instantaneous cell state is labelled by a convolutional neural network based classifier. Its feature extractor consists of five connvolutional layers with 3 × 3 kernels, each of which doubles the 3D input image in depth. Max-pooling operations downsample the activations after each convolution layer. A fully connected layer outputs a flattened array of 256 features-long representation, after which dropout is applied to perform implicit data augmentation. The final linear classifier outputs the softmax probability for each of the five classes, with the highest score considered to be the predicted label.

We performed the cell state classifier training with ~15,000 manually annotated examples of the extent of the cell nucleus, capturing the diversity of cell nuclear morphology during the cell cycle or at cell cessation. We used categorical cross entropy to calculate loss. The detailed protocol together with user-friendly tutorial to train-your-own-model for cell cycle classification is available at: https://github.com/lowe-lab-ucl/cnn-annotator.

2.3.3 Tracklet Linking Using a Bayesian Belief Matrix

The first step of the tracking algorithm is to assemble tracklets by linking cell detections over time that do not contain cell division events (**Figure 3A**).

Let $\mathcal{T}^{k-1}$ be the set of tracklets in frame $k-1$ and $\mathcal{O}^k$ be the set of cells observed in the current frame k, where track $t_j \in \mathcal{T}^{k-1}$ and object $o_i \in \mathcal{O}^k$. The algorithm attempts to calculate the posterior probability of assigning $t_j^{k-1} \rightarrow o_i^k$ and also of being lost. To do so, we create a Bayesian belief matrix (Narayana and Haverkamp, 2007) of all possible associations, $\mathbf{B}^{n\times(m+1)}$, where n is the number of active tracklets ($|\mathcal{T}^{k-1}|$), and m is the number of cell observations ($|\mathcal{O}^k|$). We initialize $\mathbf{B}$ with a uniform prior probability of association $b_{ji} = P(t_j \rightarrow o_i) = 1/(m+1)$, where

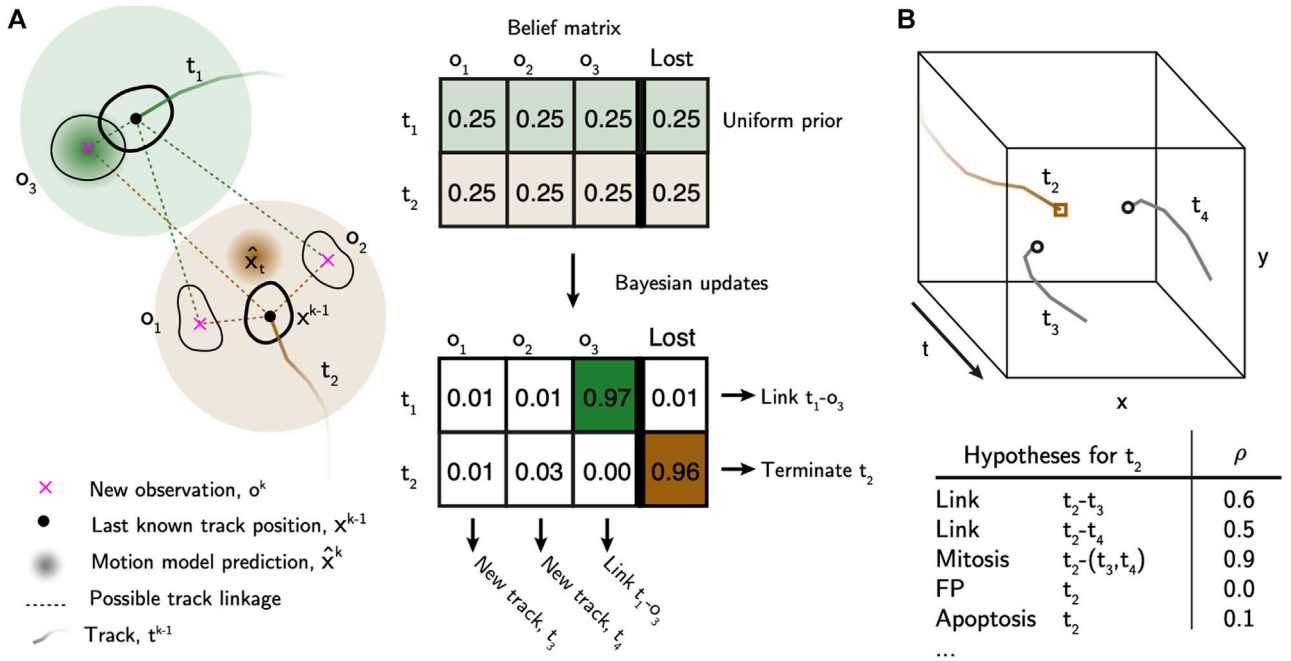


FIGURE 3 | Bayesian tracking approach overview. **(A)** Bayesian belief matrix with uniform prior is constructed for tracklet association. Using the localised and labelled cell observations, *btrack* uses Kalman filters to predict the future state of the cell ($\hat{x}$) from previous observations. Each object appearing in the new frame has uniform prior probability of track association or loss. The belief matrix is an N x (M+1) matrix where N equals number of active tracks and M is the number of detected objects per field of view. Bayesian updates are performed using the predicted track positions and state information to calculate the probability that tracks are considered lost, or to be linked to an object. Objects which are not assigned, initialise new tracks. T1 represents a simple association, where the prediction matches the observation O1. In T2 the cell divides, meaning that there is no simple association. New tracks T3 and T4 are initialised. **(B)** Global track optimisation. An n-dimensional local search creates all possible hypotheses, each with an associated likelihood, for each track based on user defined parameters. Several hypotheses are generated for Track T2, as in **(A)**: possible linkage to T3 or T4, T2 undergoes mitosis generating T3 and T4, T2 is a false positive track or that T2 undergoes apoptosis. The log-likelihood of each hypothesis (ρ) is calculated using the image and motion features. These hypotheses are evaluated in a global optimisation step to generate the final tracking results.

$m + 1$ denotes the fact that the association can be to another object, or lost.

Next, we perform Bayesian updates on **B** using evidence from motion models and the cell state classifier. Each tracklet initialises its own Kalman filter (Kalman, 1960) that is used to provide estimates of the future state of the cell. The state labels from the cell classifier are then used to associate observations to tracklets, based on visual features observed by the cell classifier which distinguishes typical nuclear morphology changes and chromatin condensation levels. For a given object o_i and a given tracklet t_j, we first update the belief matrix given the evidence E:

$$b_{ji} := \frac{P(E|t_j \rightarrow o_i) \times b_{ji}}{P(E)} \tag{1}$$

We use estimates from the Kalman filters, and a transition matrix of cell state transitions using the CNN features, to determine $P(E|t_j \rightarrow o_i)$. For the motion estimates, we use a constant velocity model in this case, although other models are possible. The output of the Kalman filter is an estimate of the future position and error in position, modelled as a multivariate normal distribution with a diagonal covariance matrix. As such, we can define a computationally simple approximation of the probability that the estimated new position of t_j lies within a units (e.g. pixels) of the new observation o_i:

$$\begin{aligned} f(x, \mu, \sigma, a) &= \frac{1}{2}\left[\mathrm{erf}\left(\frac{x - \mu + a}{\sigma\sqrt{2}}\right) - \mathrm{erf}\left(\frac{x - \mu - a}{\sigma\sqrt{2}}\right)\right] \\ P(E_{\text{motion}}|t_j \rightarrow o_i) &\approx \prod_{d \in \{x,\ y,\ z\}} f(x_d, \mu_d, \sigma_d, a) \end{aligned} \tag{2}$$

where μ_d and σ_d represent the estimate of the position and standard deviation of the motion model prediction in each spatial dimension. The advantage of this approach is that, not only do we include the difference between the new object position and the motion model prediction, but also include the motion model uncertainty into the Bayesian update. Further, we can use appearance information to inform the update. Let S be the set of states that each object is labelled with by the cell state classifier. We use a transition matrix of cell state transitions to determine the probability of linking two observations based on their respective states:

$$P(E_{\text{appearance}}|t_j \rightarrow o_i) = P(S_j = s_j, S_i = s_i),\ \ s \in S \tag{3}$$

where s_i and s_j are the states (as labelled by the cell state classifier) for the new observation and the last observation of the tracklet respectively. Importantly, this transition matrix enables the

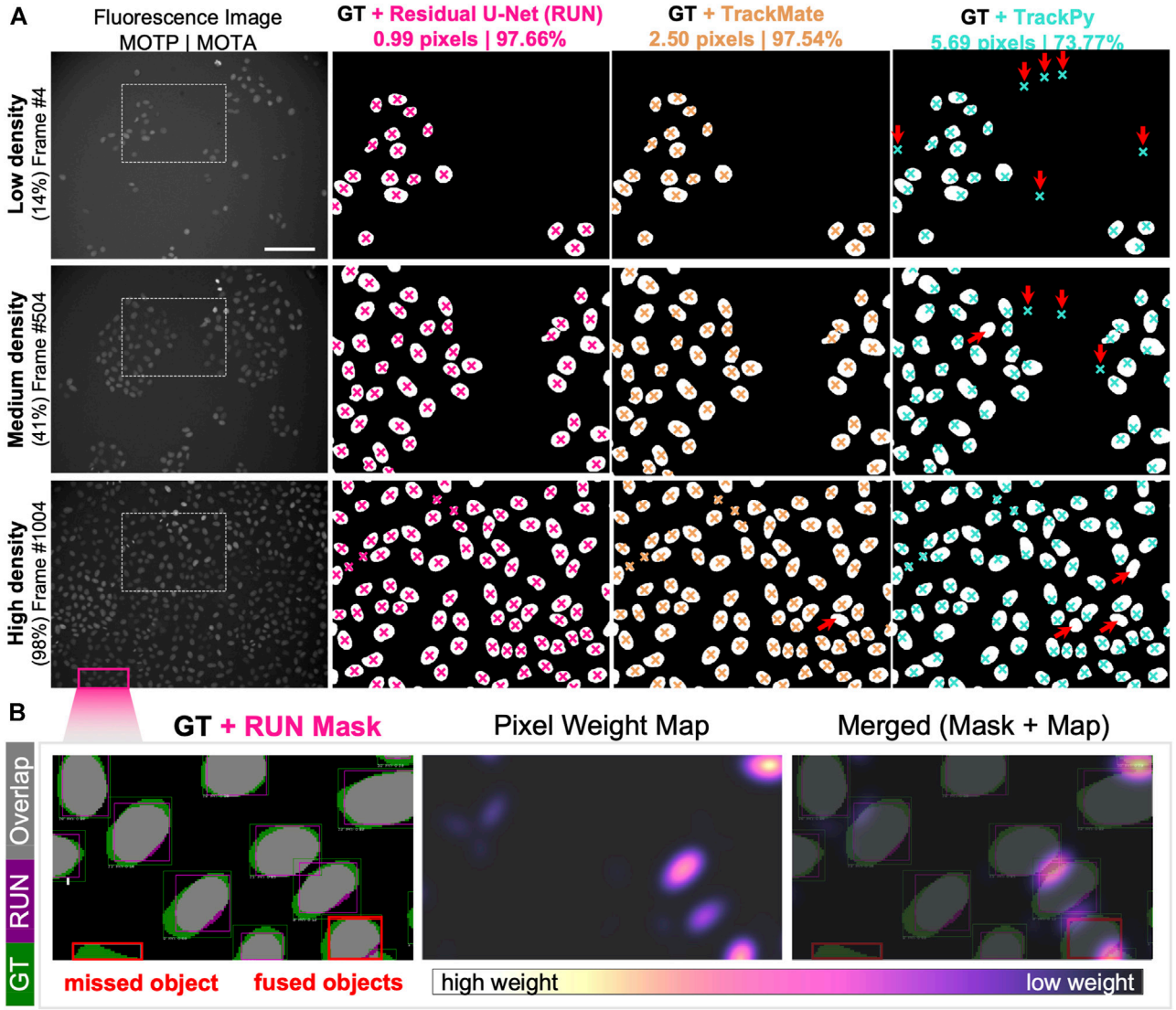


FIGURE 4 | Single-cell Segmentation Performance. **(A)** Comparison of Residual U-Net (pink), TrackMate (beige) and TrackPy (cyan) algorithms in detecting and localising cells in three representative fields of view with low, medium and high cell density. Overlaid are the cropped regions of ground truth segmentation masks with cell centroid markers depicted for zoomed-in region of the original fluorescence channel image (leftmost column). Cell localisation scores for multi-object tracking precision (MOTP) and accuracy (MOTA) are listed above. Red arrows indicate the presence of false-negative (upward right) and false-positive (downward facing arrow) cell detection errors. **(B)** Detailed overview of segmentation performance on portion of high cell density field of view. **(Left)** Per-object bounding boxes indicate visual matches between ground truth (GT, green) and Residual U-Net segmentation mask (RUN, magenta) for intersection-over-union score calculation (overlapping areas IoU, grey). Two typical segmentation error types are highlighted (red boxes), where an object is mis-detected **(left)** and two separate objects are detected as one instance **(right)**. **(Centre)** Pixel weight map designed to prioritise regions between proximal nuclei. Pixel upweighting is indirectly proportional to amount of free space between cell nuclear instances, as demonstrated in the merged image overlap **(Right)**.

tracker to penalise transitions such as Metaphase → Anaphase, preventing track linking for these cases, and allowing later steps of the algorithm to more accurately identify cell divisions. Combining these motion and appearance features, we arrive upon our estimate of $P(E|t_j \rightarrow o_i)$:

$$P(E|t_j \rightarrow o_i) = P(E_{\text{motion}}|t_j \rightarrow o_i) \cdot P(E_{\text{appearance}}|t_j \rightarrow o_i) \quad (4)$$

and:

$$P(E) = P(E|t_j \rightarrow o_i) \cdot b_{ji} + P(E|t \rightarrow o) \cdot (1 - b_{ji}) \quad (5)$$

where, $P(E|\neg t \rightarrow o)$ represents the probability of not assigning an object to a tracklet (*i.e.* the false positive detection rate), and is a hyper-parameter of the model. We then update beliefs for all other objects ($\neq i$), and the lost column, with track t_j, normalizing such that $\sum_{i=1}^{m+1} b_{ji} = 1$. As such, we treat the lost column of the belief matrix as with the real objects, and do not need to perform explicit updates.

The tracking algorithm can be used with two different methods to update the belief matrix, **B**:

- EXACT—where the updates use the full sets of objects, $\mathcal{O}^k$, which is the method of choice for all the downstream analysis in this paper, or
- APPROXIMATE—where the updates use a subset of the objects within a user defined distance from the current track. This is computationally more efficient for very large numbers of objects and tracks, at the expense of evaluating all possible linkages and penalising, for example, a mixture of fast moving objects in a field of slow moving ones.

Finally, given the belief matrix, we associate observations by choosing the hypothesis with the maximum posterior probability for each tracklet. Objects without an association initialise a new set of tracklets. Tracklets without an object association are flagged as lost and are maintained as active until reaching a threshold number of missing observations. Tracklets associated with an object are linked, and the motion and state models are updated with the new evidence. This combination of motion and cell state information in the belief matrix approach improves, for example, the subsequent detection of cell divisions, by ensuring that daughter cells are not incorrectly linked to parent cells.

2.3.4 Global Track Optimization

After linking observations into tracklets, they are next assembled into tracks and lineage trees using multiple hypothesis testing (Al-Kofahi et al., 2006; Bise et al., 2011) to identify a globally optimal solution. We built an efficient hypothesis engine, which proposes possible origin and termination fates for each track based on their appearance and motion features (**Figure 3B**, **Supplementary Materials**). The following hypotheses are generated: 1) false positive track, 2) initializing at the beginning of the movie or near the edge of the FoV, 3) termination at the end of the movie or near the edge of the FoV, 4) a merge between two tracklets, 5) a division event, or 6) an apoptotic event. In addition, we added "lazy" initialization and termination hypotheses, allowing cells to initialize or terminate anywhere in the movie. These lazy hypotheses are strongly penalized, but their inclusion significantly improves the output by relaxing the constraints on the optimization problem. The log likelihood of each hypothesis (ρ) is calculated based on the observable features and heuristics. We construct a sparse binary matrix $\mathbf{A}^{(h\times 2n)} \in \{0, 1\}$ that assigns the h hypotheses to n tracks. The matrix has $2n$ columns to account for the fact that each hypothesis may describe the initialization and termination of the same or multiple different tracklets. For example, in the case of track joining between tracklets t_i and t_j, a 1 is placed in columns i and $n + j$. In the case of a mitotic branch, a 1 is placed in column i for the parent tracklet t_i, and columns $n + j$ and $n + k$ for the two child tracklets t_j and t_k. As such, the matrix **A** can be used to account for all tracklets in the set. We then solve for the optimal set of hypotheses (x^*) that maximises the likelihood function:

$$x^* = \underset{x}{\operatorname{argmax}} \rho^\top x, \text{ s.t. } \mathbf{A}^\top x = 1 \tag{6}$$

where the optimization space of x is also a binary variable ($x \in \{0, 1\}$). As a result, the optimal set of hypotheses (x^*) accounts for all tracklets in the set. Once the optimal solution has been identified, tracklet merging into the final tracks can be performed. A graph-based search is then used to assemble the tracks into lineage trees, and propagate lineage information such as generational depth, parent and root IDs, from which the tree root (founder cell identity) and leaf cells (those with no known progeny) can be identified.

3 RESULTS

To assess the heterogeneity and the heritability of cell-cycle duration in a population of cells, we sought to automatically reconstruct family trees from individual cells in long-term time-lapse movies. Our time-lapse microscopy dataset (see Section 2.1) spans densities from single cells to highly confluent monolayers, providing a unique dataset to thoroughly test the computational framework.

To provide an initial baseline for our pipeline's performance on our non-trivial imaging dataset, the assessment of our movie analysis workflow was contrasted to 1) a Python-based, general-purpose particle tracking package, TrackPy (Allan et al., 2018), and 2) a cell tracking-specific ImageJ/FiJi plugin, TrackMate (Tinevez et al., 2017). Both of these tracking engines represent recently-developed popular cell and/or particle tracking frameworks which function as backbones to the current state-of-the-art cell tracking software, such as Usiigaci (Tsai et al., 2019) and MaMuT (Wolff et al., 2018) or Mastodon (Tinevez, 2021), respectively. Detailed description of the tracking softwares calibration can be found in the *Supplementary Materials* section.

3.1 Bayesian Cell Tracking Approach

To automate data analysis, we developed a fast, open source and easy-to-use cell tracking library to enable calculation of intermitotic durations and the capture of multi-generational lineage relationships. Our pipeline consists of three steps: 1) cell segmentation, 2) cell state labelling, where progression of cells towards division is classified, and 3) cell tracking with lineage tree reconstruction.

3.1.1 Cell Detection Performance

To assess the quality of the cell detection step, we selected three representative fields of view capturing fluorescently labelled nuclei of cells at low, medium and high confluency. We manually annotated the nuclear areas by circling around the cell nuclei to create an instance segmentation mask where pixels of value 0 = background, 1 = foreground, i.e. individual cells. We then computed the nuclear centroid of each cell instance and measured the fidelity of cell detection and localisation by calculating multi-object tracking precision (MOTP, **Supplementary Equation S1**). Out of 869 cells in total, 847 cells had their centroid localized within acceptable error of ≈1 nuclear radius length (<20 pixels ≈ 7 μm) compared to the ground truth annotations. We report that our pipeline reached sub-pixel precision and outperforms both particle detection

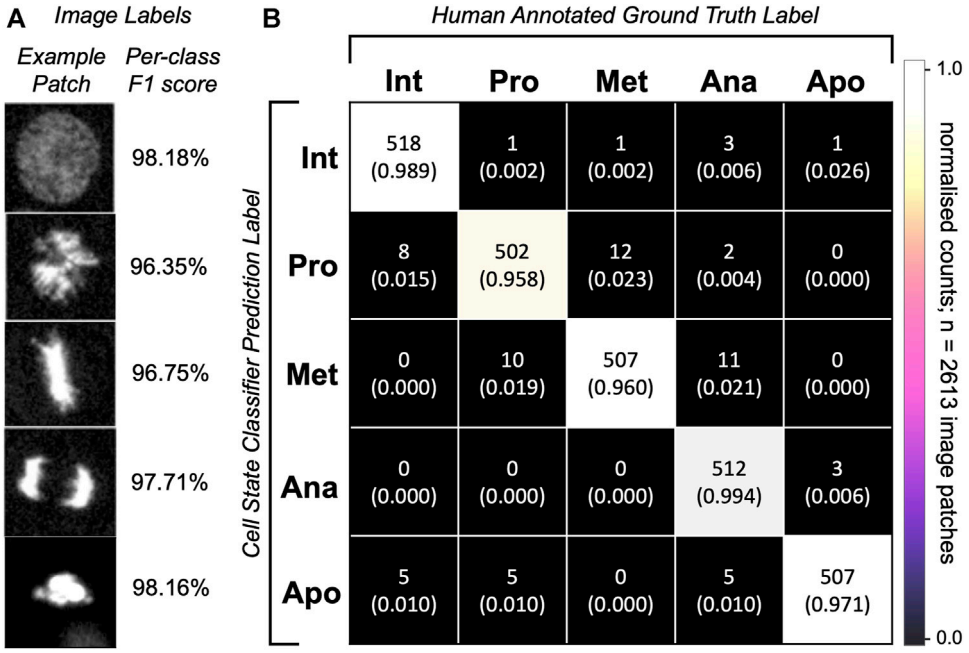


FIGURE 5 | Single-cell classification performance. **(A)** Representative predictions from the trained network for the given images patches. Depicted is an image patch (fluorescence channel only) centred at the cell nucleus representative of the class label with a per-class F1-score calculated on 2,613 single-cell images in the testing dataset. **(B)** Confusion matrix for cell state labelling of the testing patch dataset (>500 images per class) of cell image crops. Per-class image patch counts and their normalised counts are depicted. Comparison of human annotations vs. the classifier-generated annotations are shown.

approaches with > 5-fold enhancement of the TrackPy- and >2.5-fold higher localisation precision than TrackMate-embedded cell detection method (**Figure 4A**).

Our cell localisation approach performs a pixel-wise image classification, which offers the opportunity to segment the whole area belonging to the cell nucleus from which centroid coordinates are calculated. To calculate the accuracy of the nuclear area segmentation, we computed the per-object metrics from the residual U-Net and compared them to the manually labelled ground truth (**Figure 4B**). We report that out of 616 ground truth nuclei in a nearly confluent FoV (**Figure 4A**), 574 nuclei had at least 50% Intersection over Union (IoU, **Supplementary Equation S2**), yielding a Jaccard Index (*J*, **Supplementary Equation S3**) of 0.933, meaning that 93% of all objects in the FoV were correctly detected. Across the total of 869 cells detected in all assessed FoVs, we achieved an IoU equal to 0.802 and a *J* of 0.975, while retaining a pixel identity (*PI*, **Supplementary Equation S4**) at 0.874 (**Supplementary Materials**).

To assess the cell state classification performance, we calculated a harmonic mean between the precision and recall (F1-score, **Supplementary Equation S5**) for each class to be distinguished by our model classifier (**Figure 5A**). To do this, we plotted a confusion matrix of >2,600 single cell image patches and compared the classifier predictions with human annotations (**Figure 5B**). Our optional cell state labelling step yields an overall accuracy of >97% with all state labels classified with >96% F1-scores (**Figure 5A**).

3.1.2 Trajectory Following Accuracy

Next, we calculated the multi-object tracking accuracy (MOTA, **Supplementary Equation S6**) which scores the tracker's ability to retain cell's identity and trajectory over longer periods of time. The MOTA score intrinsically penalises the tracking pipeline for static (falsely positive and falsely negative detections) as well as dynamic (identity switches) errors, which often strongly rely on the detection algorithm performance.

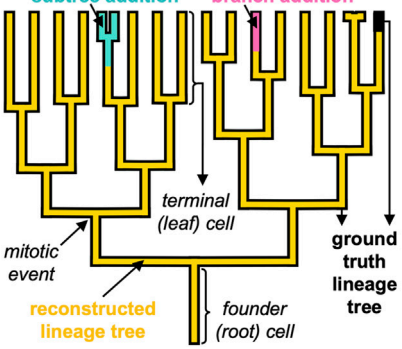
We used short movie sequences of up to 20 successive frames from a FoV at three different cell density levels. Observing 2,161 individual cell objects over time, our tracking pipeline performed with 97.66% accuracy of between-frames associations. Identical dataset assessed by TrackMate resulted in a comparable score of 97.54%, while tracking by TrackPy yielded 73.77% accuracy. These results confirm that tracklet linking and object associations between subsequent frames are performed with comparably high fidelity across both tracking tools designed specifically for cell tracking.

However, because the MOTA score is not designed to detect track splitting events, we sought alternative metrics to assess the fidelity of cell division history recording. Although multiple approaches exist to cross-compare tracking performances (Ulman et al., 2017), few of those look specifically at the assessment of branching events of parent tracks and correct assignment of generational depth to the resulting children tracks. Below, we describe the adopted as well as newly defined metrics for measuring the fidelity of track splitting events. Although we continued with using TrackMate for

A

Cell Division Metrics	Ground Truth Event Count	*BayesianTracker*		*TrackMate*	
		With Penalty	Penalty Free	With Penalty	Penalty Free
Mitotic Branching (MBC)	351 mitoses	86.76%	87.21%	42.65%	26.13%
Leaf Retrieval (LRS)	364 leaf cells	78.84%	84.05%	14.44%	29.69%

B



C

Validation of Lineage Tree Reconstruction

Manual Tree Curation	Total Cells Validated	Broken Tracks	Frequency [in frames]
Branch breakage	728	25	1 in ~6,000
Re-assembly actions	1 subtree swap	8 subtrees	25 branches
Frame additions		+ 11,505	+ 1,926
Lineage Tree Fidelity	**No curation**	**+ Subtrees**	**+ Branches**
Precision (specificity)	99.72%	99.74%	99.75%
Recall (sensitivity)	90.55%	98.88%	99.83%

FIGURE 6 | Summary of Benchmarking Metrics to Evaluate the Tracking Performance. **(A)** Summary of cell division-specific metrics for tracker benchmarking on identical number of ground truth events. **(B)** Visual overview of automated tree reconstruction with additional manual tree re-assembly. Highlighted are the regions of ground truth trees (black thick background) which were correctly recapitulated by *btrack* (overlaid golden lineage). Upon branch breakage, two types of assembly actions were applied: subtree attachment (cyan), where the branch underwent further splitting, or branch attachment (pink), where the track did not further branch. **(C)** Summary of additional benchmarking metrics to score the performance of *btrack* approach with respect to single-cell trajectory following and lineage tree reconstruction in naïve *btrack* outputs and expert annotator re-assembled trees.

additional comparisons on our validation dataset (**Supplementary Materials**), it should be noted that further analysis by TrackPy was excluded as this tracking algorithm currently lacks the capacity to follow cases of track branching, typical for dividing cells.

3.1.3 Lineage Tree Reconstruction Fidelity

Reconstructing lineages represents a complex challenge and a common source of tracking errors in multigenerational cell observations. From our track validation dataset, we automatically reconstructed 154 lineage trees which contained at least one mitotic event. These trees were initiated in three instances: 1) through their initial presence in the FoV at the beginning of a movie, 2) as a consequence of migration into the FoV, or 3) upon breakage of an existing branch from its tree.

We manually reconstructed the ground truth lineage trees of 24 randomly selected founder cells, the subsequent progeny of which spanned the entire movie duration and accounted for 1,032 cells including tree founders (**Supplementary Materials**). This sample accounted for more than 1/3 of initially seeded cells and served as validation dataset for comparison of the ground truth observations to their respective automatically reconstructed lineages. We subjected these trees to multiple lineage fidelity metrics for benchmarking to robustly assess the cell tracking performance with and without manual curation.

For ease of lineage tree visualisation, we provide a convenient visualisation for track survival over time (**Supplementary Movie S1**), revealing trajectories that can be tracked all the way to the movie start *via* their ancestors, as an easy way to visually demonstrate the tracking fidelity and identify errors.

3.1.4 Cell Division Correctness

To measure the correctness of parent-child relationships, we calculated the mitotic branching correctness score (MBC, **Supplementary Equation S7**; **Figure 6A** and **Supplementary Materials**) for the detection of mitosis. Here, cell divisions are correctly identified when computer-identified mitoses can be mapped to human-annotated ground truth lineage trees (Bise et al., 2011). Additionally, we introduced a penalty for excessive branching assignment where cells were only scored as "hits" when they were assigned the correct generational depth relative to their founder cell. Similarly, we defined a leaf retrieval score (LRS, **Supplementary Equation S8**; **Figure 6A** and **Supplementary Materials**), which computes the number of all terminal cells in the respective lineage tree appearing in the last frame of the movie. To account for the fidelity of generational depth, in the penalty-applied MBC and LRS calculations, the progeny were considered as correct only when the generational depth relative to the tree root matched the ground truth (**Supplementary Materials**).

Keeping record of cell division history over the entire duration of the live-cell imaging is necessary to elucidate whether cells with shorter cell cycles have the capacity to dominate the population and competitively outgrow the slowly dividing cells over time. Our software correctly records the cell division history as reflected by both MBC and LRS scores, >86% with and >87% without penalty for MBC and >78% with and >84% without penalisation for LRS

(**Figure 6A**). We report that both the MBC and LRS scores were considerably higher for our *btrack* algorithm with between ~ 2-fold and 5.5-fold better performance than our benchmarking standard using both penalised and penalty-free variants (**Figure 6A**), respectively.

These results highlight the utility of the extra labelling information for cell tracking, as well as our tracker's sophistication upon building complex predictions based on position and state of each cell. This is in contrast to approaches which employ hard distance thresholding to account for splitting events, or which minimise link distances by connecting cells with their closest neighbour in the subsequent frames. Such algorithms are by default unable to discriminate between false cell divisions, where two independent cells migrate into close proximity below set threshold, and true mitotic events which are accompanied by characteristic morphological changes in nucleic acid organisation in the parent dividing cells as well as progeny cells. Accounting for the "metaphase" and "anaphase" labels in our approach leads to minimising of the occurrence of parent-to-child track fusions (**Supplementary Materials**). Our scoring system therefore manifests the importance of our optional cell state classification step, which generates additional information for correct division hypotheses creation utilised by the global optimisation engine on the tracking data.

3.1.5 Manual Tree Re-Assembly

To compute further metrics for assessing *btrack* performance in multi-generational lineage reconstruction, we performed manual tree re-assemblies of our automated *btrack* output data, where necessary (**Figure 6B**, **Supplementary Materials**). This allowed us to perform detailed error analysis by aligning re-assembled trees to the ground truth observations and evaluate the overall extent of the need for human lineage curations, discussed next.

Branch breakages occur due to incorrect cell segmentation (e.g. due to low fluorescence signal), or through tracking errors. However, these breakage errors were very rare (one breakage per ~6,000 correctly linked frame observations, **Figure 6C**). When the lost cell is re-detected, it becomes the founder of a newly initialised track. This track now becomes the root of a new tree, which can continue to further grow and divide as a "subtree" of the true tree or exist without further splitting as "branch" (**Figure 6B**).

3.1.6 Tracking Effectiveness

Finally, we computed metrics reflecting how well the human-annotated cell trajectories are followed by the reconstructed lineage trees (recall; **Supplementary Equation S9**) and vice versa (precision; **Supplementary Equation S10**) (Bise et al., 2011).

Without any manual curation, our *btrack* algorithm faithfully follows >90% of the human-annotated cell trajectories (recall) with over >99% of observations in agreement with the ground truth trees (precision) (**Figure 6C**). Taking further advantage of our large dataset, we randomly sampled an additional 302 unseen, automatically reconstructed trees from multiple movies. For each tree, we visually scored them as sufficient if they had >90% recall of the ground truth tree as determined by an expert annotator, which added up to 201 out of 370 reconstructed trees overall.

To quantify the remainder of the trees containing one or more errors, we returned to our fully annotated validation dataset to manually re-assemble the broken trees. We needed to perform only eight "subtree" re-assembly actions to the original trees (**Supplementary Materials**), which added 11,505 frame observations, and a single subtree swap (where part of the tree was falsely associated with another tree). Although the accuracy scores were further enhanced to >99% recall by additional human annotation *via* tree re-assembly (**Figure 6B**), we report that our workflow achieves competitive performance on long-term live-cell imaging data even in fully unsupervised manner (**Figure 6C**). This is essential to investigate the existence of inheritable cell cycle characteristics and their regulation using large datasets.

3.2 Single Cell Cycling Duration Analysis

Having validated the lineage tree reconstructions, we pooled the tracking data from the entire dataset of 44 time-lapse movies. First, we removed those cells which were partially resolved i.e. founding parents (root cells) or progeny (leaf cells) in the reconstructed lineages. This yielded 22,519 cells, organised into lineage trees spanning up to eight generations. Next, we calculated the per-cell inter-mitotic time as the time between the first appearance of separated chromosomes during mitosis (labelled as anaphase by the CNN; due to the temporal sampling this may be before cytokinesis occurs) to the frame preceding the next anaphase (**Figure 7A**).

Detailed inspection of the nuclear areas and CNN labels lead us to exclude certain cells from our distribution. We found that the tracks with inter-mitotic time below 7 h had high incidence of start with a non-anaphase label, did not end with pro-(meta) phase label, or a combination of both. This observation suggested that these short tracks represent fragments of cell trajectories where branch breakages occurred, rather than being representative of ultra-fast cycling cells. Visual observation of the nuclear growth (increase of cell nucleus segmentation mask area over time) indicated that tracks with cycling time over 42 h often captured track instances where a parent cell (undergoing mitosis) was falsely linked to one of the arising children cells, most likely due to imperfection in the segmentation step.

3.2.1 Heterogeneity in Cell Cycling Durations

To avoid possible incorporation of prematurely terminated tracks, concatenated parent-to-child tracks or other tracking errors, we filtered our pooled dataset to only consider cells with cycling lengths between and including 7 and 42 h for further analysis (**Figure 7B**). Our final dataset consisted of 20,074 cells with known lineage over up to eight generations, representing at least two orders of magnitude greater numbers than in previous studies (Puliafito et al., 2012). Importantly, the whole process to filter the relevant cells, calculate their division times and pool the single-cell information across 44 movies took less than a minute to produce, illustrating the utility of our tracking pipeline for large data analyses.

The pooled dataset shows a positively skewed normal distribution of cell cycle durations (**Figure 7B**). This

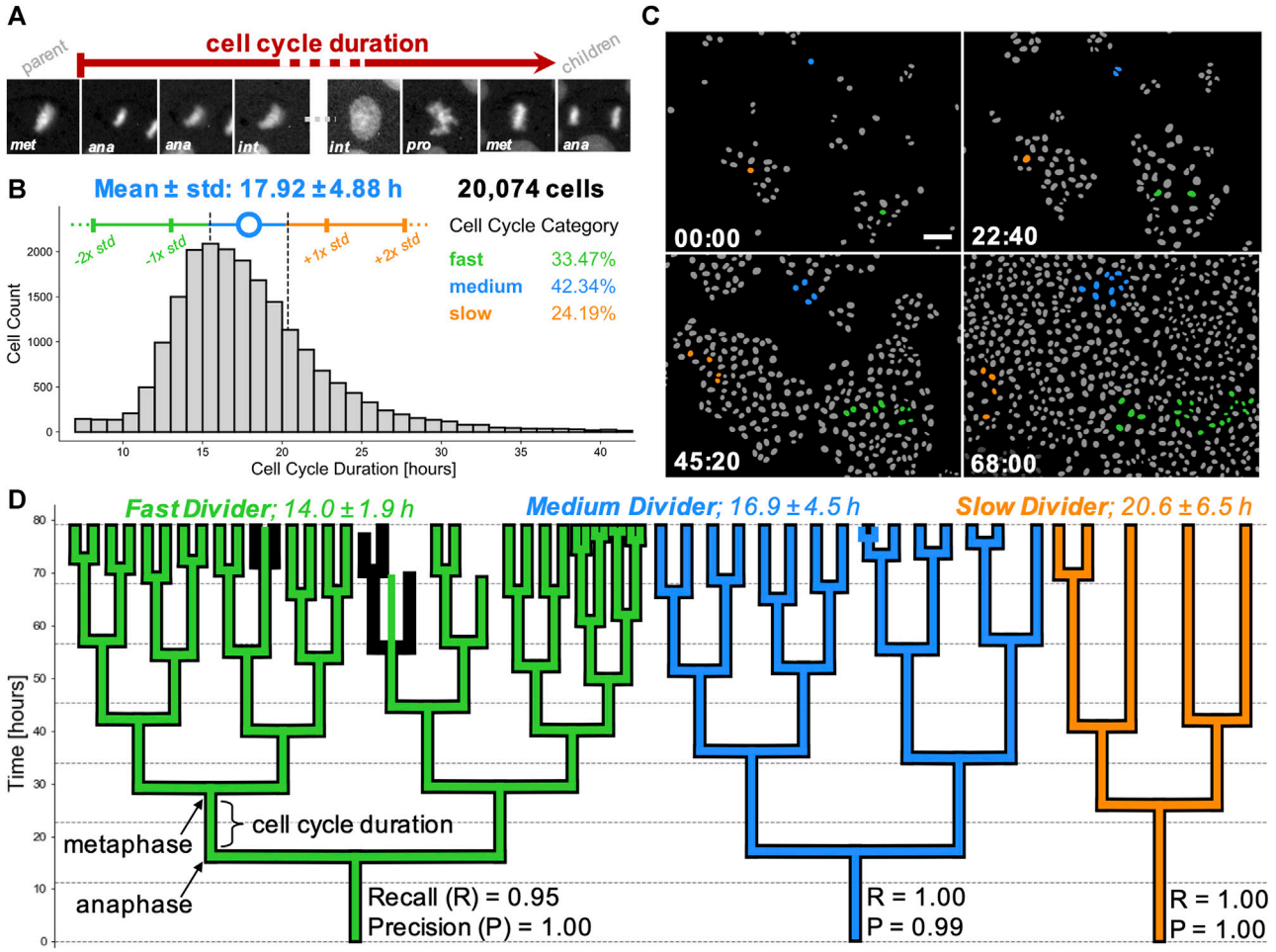


FIGURE 7 | Cell cycling heterogeneity and colony expansion capacity within single cell clones. **(A)** Cell cycle duration, defined here as the time elapsed between two subsequent anaphases, from the mitosis of reference cell's parent (first image patch) to division into two children cells (last image patch). Scale bar = 5 μm. **(B)** Calculated cell cycle duration distribution over the pooled MDCK live-cell imaging dataset from 5,325 unique lineage trees. Error bars (colored vertical bars) show one and two standard deviations around the mean (blue circle) of tracked cells for generational depths spanning 1 to 6 (generations #0 and #7 were excluded as those corresponded to the tree root of leaf cells). Cell categorisation into fast (green), medium (blue) and slow (orange) dividers is depicted as below, within or above 1/2 standard deviation away from the mean (dashed black vertical lines), respectively. Percentages of single cells belonging to each category are stated. **(C)** Sequence of four colourised binary masks with segmented individual cells (grey) on background (black), highlighting cell proliferation from the start (top left) to the end (bottom right) of a representative movie. Time in hh:mm is indicated in the bottom left corner. Founder cells and progeny corresponding to slow (orange), medium (blue) and fast (green) dividers are highlighted. Scale bar = 50 μm. **(D)** 2D representations of typical lineage trees captured from data, showing slow (orange), medium (blue) and fast (green) dividing cells. Computer-generated tracks (colour) are overlaid on human-reconstructed ground truth trajectories (black). Individual recall and precision scores are shown for each lineage tree, with vertical branch lengths corresponding to intermitotic time (cell cycle duration) elapsed between the first post-division frame (anaphase) to the last frame prior to the next cell division (metaphase). Trees illustrate an error-free tracking (orange tree) and two types of tracking errors, i.e. falsely identified mitosis (blue tree) and missed mitotic event (green tree).

distribution confirms the presence of division time heterogeneity at very large population numbers, while standing in good agreement with previously published work (Puliafito et al., 2012; Bove et al., 2017). We also confirmed that cell cycle length was not correlated with the time at which a cell was born for the first 60 h of time-lapse imaging (**Supplementary Materials**).

We used the population distribution to categorise individual cells as fast, medium and slow dividers based on whether their cycling duration was below (<15.5 h), within or above (>20.3 h) half a standard deviation away from the sample mean (**Figure 7B**). This definition yields a subpopulation of fast dividing cells (n = 6,719 cells; 33.5% of the population), medium dividers (n = 8,500 cells; 42.3%) and slow dividers (n = 4,855 cells; 24.2%).

We extended this categorisation to describe individual cell families and contrasted three lineages extracted from our representative movie, with the trees observed from the movie start frame. Mapping the number of progeny originating from these three founder cells (**Figure 7C**, **Supplementary Movie S2**), we confirmed our large-scale observation of broad cycling heterogeneity spectrum. Based on the average cell cycle length

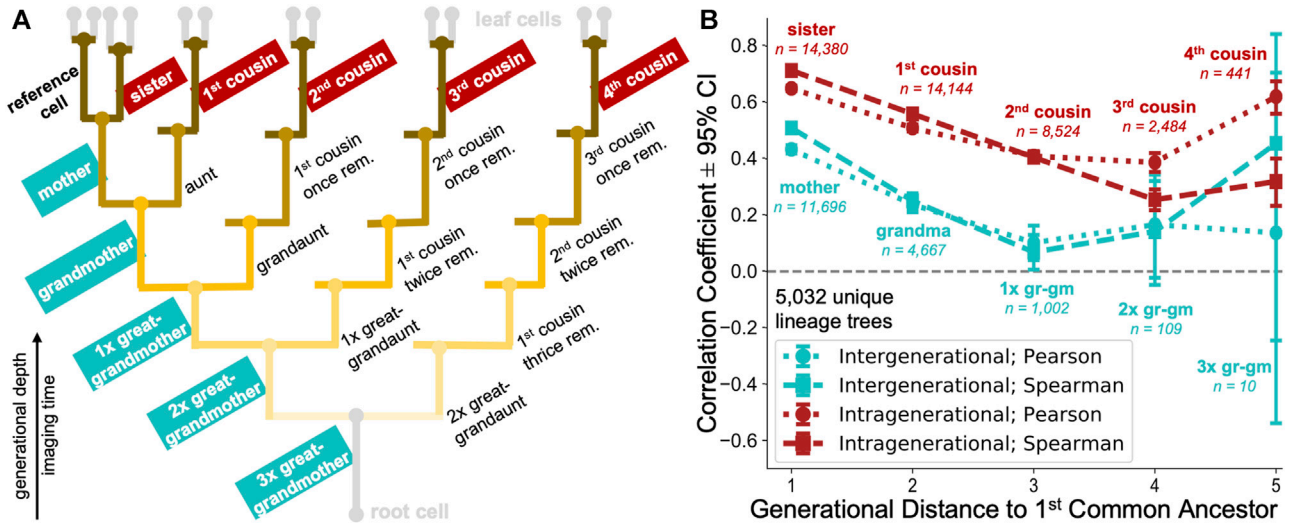


FIGURE 8 | Large-scale multigenerational analysis of single-cell cycling durations. **(A)** Illustrative lineage tree showing 20 types of family relationships of a reference cell (bold) to its lineage relatives determined using our automated approach. The tree captures extended kinships on ancestral (inter-generational; generationally unequal) to fourth cousin span (intra-generational; generationally equal). Strength of the branch colour shading illustrates the generational distance to nearest common ancestor between family member and reference cell. **(B)** Cycling length correlations in cell lineages. intra-generational (red) correlations appear to be consistently higher than the inter-generational (cyan) correlations in directly ancestral branches. Pearson (dotted) and Spearman (dashed) rank correlation coefficients with 95% confidence interval (CI) are shown for selected relationship pairs as in the lineage tree schematic, categorised according to the generational distance to first common ancestor.

of all fully-resolved cells within the tree, we classified the cell families as slow, medium and fast cyclers. Visual inspection of their lineage tree representations reveals that variability in intermitotic durations amongst single cells directly influences the cell capacity to divide and raises the potential for fast cycling cells to eventually dominate the population by overgrowing the slower-cycling clones (**Figures 7C,D**), as suggested previously (Mura et al., 2019).

Indeed, taking trees corresponding to three different root cells, we find that at the end of the movie, the slow dividing family results in five leaf cells, the medium dividing family in 15 cells and the fast dividing family in 31 leaf cells, over the same period of 80 h. These correspond to mean intermitotic durations of 20.6 ± 6.5 h (n = 3 cells), 16.9 ± 4.5 h (n = 14 cells), and 14.0 ± 1.9 h (n = 29 cells) for each tree, respectively (**Figure 7D**). Our findings suggest a high degree of intrinsic cell cycling heterogeneity present in the wild-type MDCK cell population. This cycle-time heterogeneity appears to be maintained within cell lineages.

3.2.2 Cell Cycle Duration Correlations in Deep Lineages

To study whether the observed cell cycle duration could represent an inheritable characteristic, we extracted 20 different types of cell pair relatedness from up to 8-generation deep lineages (**Figure 8A**). It was previously observed (Sandler et al., 2015) that cell cycle durations within lineages show poor correlation when observing the directly ancestral, inter-generational cell pairs (mother-daughter and grandmother-granddaughter), but remain highly correlated when examining intra-generational relationships (sister and cousin cells) (Sandler et al., 2015; Chakrabarti et al., 2018; Kuchen et al., 2020). However, due to lack of deep datasets, the analysis has traditionally focused on immediate cell relatives, including mother, grandmother, sister and first cousin relationships (Sandler et al., 2015; Chakrabarti et al., 2018) and less frequently on more remote family members, such as great-grandmothers or second cousins (Kuchen et al., 2020).

Enabled by our automated approach, we calculated the correlation between cell cycle duration in lineages captured for >8 generational depths, extending previous studies (Sandler et al., 2015; Chakrabarti et al., 2018; Kuchen et al., 2020) using the enhanced depth, breadth and number (5,032 trees) of automatically reconstructed lineages in our tree pool. If the cell cycle duration is not controlled in a heritable fashion, we would expect poor cycling correlations across inter-generational (generationally unequal) as well as intra-generational (generationally equal) family relationships (**Figure 8A**).

Our analysis revealed moderate correlations in immediate inter-generational relationships (Pearson and Spearman rank coefficients of 0.43 and 0.51, respectively between 11,696 reference and mother cell pairs), which rapidly decreased over 2–3 rounds of division (**Figure 8B**). Such diminishing ancestral correlations would be strongly indicative of cell cycle duration inheritance being a stochastic event, as previously speculated (Sandler et al., 2015). Alternatively, this behaviour could also indicate that the accumulation of noise across inter-generational lineage relationships represents a very rapid process.

Our analysis confirmed a previously observed trend of highly synchronised cell cycling durations in sister cell pairs (Pearson and Spearman rank coefficients of 0.65 and 0.71, respectively between 14,380 sister cells examined). In addition, our analysis revealed that intra-generational correlations remained above 0.5 in 1st and 2nd cousins. Over five generations, intragenerational correlations became progressively less marked but remained consistently larger than inter-generational correlations.

Our results derived from high-replicate lineage data are in good agreement with previously published studies, using manually annotated data (Sandler et al., 2015; Chakrabarti et al., 2018; Kuchen et al., 2020). Overall, the data provide additional evidence that cycling length is correlated across longer-range relationships (4 and 5 generations to nearest common ancestor) than previously examined. These long-term correlations may suggest heritability in cell cycle durations, as proposed by others (Mura et al., 2019). However, it is possible that, as culture conditions evolve with time (nutrients become depleted and cells more confluent), the highly correlated behaviour between same generation family members represents a consequence of environmental synchronisation. Further experiments are needed to discriminate between these hypotheses.

4 DISCUSSION

We developed an easy-to-use, open-source Python package to enable rapid and accurate reconstruction of multi-generational lineage trees from large datasets without time-consuming manual curation. We show that the addition of cell state information into a probabilistic single cell tracking framework improves multi-generational lineage tree reconstruction over other approaches. Our software enables users to characterise population-level relationships with single-cell resolution from time-lapse microscopy data in an unsupervised manner. As a demonstration of our fully automated approach, we extend the cell cycling analysis to family relationships which could not be previously examined in (semi-) manual annotation-dependent studies (Sandler et al., 2015; Chakrabarti et al., 2018; Kuchen et al., 2020) due to lack of experimental data. We envisage our tracking software could further be applied to analysis of (cancer) stem cell identification in tissues, detection of differentiation and/or reprogramming success in populations, studying of cell cycle control mechanisms in cell lineages and dynamic high-throughput screens of various pharmaceutical compounds with live-cell imaging.

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

GC and AL conceived and designed the research. KU and GV performed experiments. KU developed and performed computational analysis. AL wrote the image processing and cell tracking code. KU, GV, GC, and AL evaluated the results and wrote the paper.

FUNDING

KU was supported by a BBSRC LIDo studentship. GV was supported by BBSRC grant BB/S009329/1 to AL and GC.

ACKNOWLEDGMENTS

We thank members of the AL and GC labs for helpful discussions. We thank the reviewers for their time and insightful comments.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2021.734559/full#supplementary-material

Supplementary Video S1 | https://youtu.be/ZxywQ7Laihl

Supplementary Video S2 | https://youtu.be/gScvX89JeYQ

REFERENCES

Akram, S. U., Kannala, J., Eklund, L., and Heikkilä, J. (2017). *Cell Tracking via Proposal Generation and Selection*, *CoRR* abs/1705, 03386.

Al-Kofahi, O., Radke, R. J., Goderie, S. K., Shen, Q., Temple, S., and Roysam, B. (2006). Automated Cell Lineage Construction: A Rapid Method to Analyze Clonal Development Established with Murine Neural Progenitor Cells. *Cell Cycle* 5, 327–335. doi:10.4161/cc.5.3.2426

Allan, D. B., Caswell, T., Keim, N. C., and van der Wel, C. M. (2018). *Trackpy: Trackpy v0.4.1*. doi:10.5281/ZENODO.1226458

Amat, F., Lemon, W., Mossing, D. P., McDole, K., Wan, Y., Branson, K., et al. (2014). Fast, Accurate Reconstruction of Cell Lineages from Large-Scale Fluorescence Microscopy Data. *Nat. Methods* 11, 951–958. doi:10.1038/nmeth.3036

Bannon, D., Moen, E., Schwartz, M., Borba, E., Kudo, T., Greenwald, N., et al. (2021). DeepCell Kiosk: Scaling Deep Learning-Enabled Cellular Image Analysis with Kubernetes. *Nat. Methods* 18, 43–45. doi:10.1038/s41592-020-01023-0

Bao, Z., Murray, J. I., Boyle, T., Ooi, S. L., Sandel, M. J., and Waterston, R. H. (2006). Automated Cell Lineage Tracing in *Caenorhabditis elegans*. *Proc. Natl. Acad. Sci.* 103, 2707–2712. doi:10.1073/pnas.0511111103

Bendall, S. C., Davis, K. L., Amir, E.-a. D., Tadmor, M. D., Simonds, E. F., Chen, T. J., et al. (2014). Single-cell Trajectory Detection Uncovers Progression and Regulatory Coordination in Human B Cell Development. *Cell* 157, 714–725. doi:10.1016/j.cell.2014.04.005

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., et al. (2019). Ilastik: Interactive Machine Learning for (Bio)image Analysis. *Nat. Methods* 16, 1226–1232. doi:10.1038/s41592-019-0582-9

Bise, R., Yin, Z., and Kanade, T. (2011). "Reliable Cell Tracking by Global Data Association," in 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Chicago, IL, USA, 30 March-2 April 2011, 1004–1010. doi:10.1109/ISBI.2011.5872571

Bove, A., Gradeci, D., Fujita, Y., Banerjee, S., Charras, G., and Lowe, A. R. (2017). Local Cellular Neighborhood Controls Proliferation in Cell Competition. *MBoC* 28, 3215–3228. doi:10.1091/mbc.E17-06-0368

Chakrabarti, S., Paek, A. L., Reyes, J., Lasick, K. A., Lahav, G., and Michor, F. (2018). Hidden Heterogeneity and Circadian-Controlled Cell Fate Inferred from Single Cell Lineages. *Nat. Commun.* 9, 1–13. doi:10.1038/s41467-018-07788-5

Dagogo-jack, I., and Shaw, A. T. (2017). Tumour Heterogeneity and Resistance to Cancer Therapies. *Nat. Rev. Clin. Oncol.* 15, 81–94. doi:10.1038/nrclinonc.2017.166

Downey, M. J., Jeziorska, D. M., Ott, S., Tamai, T. K., Koentges, G., Vance, K. W., et al. (2011). Extracting Fluorescent Reporter Time Courses of Cell Lineages from High-Throughput Microscopy at Low Temporal Resolution. *PLoS ONE* 6, e27886. doi:10.1371/journal.pone.0027886

Edelstein, A. D., Tsuchida, M. A., Amodaj, N., Pinkard, H., Vale, R. D., and Stuurman, N. (2014). Advanced Methods of Microscope Control Using μManager Software. *J. Biol. Methods* 1, 10. doi:10.14440/jbm.2014.36

Faure, E., Savy, T., Rizzi, B., Melani, C., Stašová, O., Fabrèges, D., et al. (2016). A Workflow to Process 3D+time Microscopy Images of Developing Organisms and Reconstruct Their Cell Lineage. *Nat. Commun.* 7. doi:10.1038/ncomms9674

Fazeli, E., Roy, N. H., Follain, G., Laine, R. F., Chamier, L. V., Hänninen, P. E., et al. (2021). *Automated Cell Tracking Using StarDist and TrackMate [Version 1 ; Peer Review : 2 Approved , 1 Approved with Reservations*, 1–13.

Fazeli, E., Roy, N. H., Follain, G., Laine, R. F., von Chamier, L., Hänninen, P. E., et al. (2020). *Automated Cell Tracking Using Stardist and Trackmate*. Cold Spring Harbour Laboratories. doi:10.1101/2020.09.22.306233

Han, H., Wu, G., Li, Y., and Zi, Z. (2019). eDetect: A Fast Error Detection and Correction Tool for Live Cell Imaging Data Analysis. *iScience* 13, 1–8. doi:10.1016/j.isci.2019.02.004

He, K., Zhang, X., Ren, S., and Sun, J. (2015). *Deep Residual Learning for Image Recognition*, 03385. CoRR abs/1512.

Hernandez, D. E., Chen, S. W., Hunter, E. E., Steager, E. B., and Kumar, V. (2018). "Cell Tracking with Deep Learning and the Viterbi Algorithm," in 2018 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS), Nagoya, Japan, 4-8 July 2018, 1–6. doi:10.1109/MARSS.2018.8481231

Hilsenbeck, O., Schwarzfischer, M., Skylaki, S., Schauberger, B., Hoppe, P. S., Loeffler, D., et al. (2016). Software Tools for Single-Cell Tracking and Quantification of Cellular and Molecular Properties. *Nat. Biotechnol.* 34, 703–706. doi:10.1038/nbt.3626

Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S. L., et al. (2008). Robust Single-Particle Tracking in Live-Cell Time-Lapse Sequences. *Nat. Methods* 5, 695–702. doi:10.1038/nmeth.1237

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* 82 (1), 35–45. doi:10.1115/1.3662552

Kingma, D. P., and Ba, J. L. (2015). "Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations," in ICLR 2015 - Conference Track Proceedings, San Diego, 2015, 1–15.

Kuchen, E. E., Becker, N. B., Claudino, N., and Höfer, T. (2020). Hidden Long-Range Memories of Growth and Cycle Speed Correlate Cell Cycles in Lineage Trees. *eLife* 9, e51002. doi:10.7554/eLife.51002

Lugagne, J.-B., Lin, H., and Dunlop, M. J. (2020). DeLTA: Automated Cell Segmentation, Tracking, and Lineage Reconstruction Using Deep Learning. *Plos Comput. Biol.* 16 (4), e1007673. doi:10.1371/journal.pcbi.1007673

Magnusson, K. E. G., Jaldén, J., Gilbert, P. M., and Blau, H. M. (2015). Global Linking of Cell Tracks Using the Viterbi Algorithm. *IEEE Trans. Med. Imaging* 34, 911–929. doi:10.1109/TMI.2014.2370951

Mandal, S., and Uhlmann, V. (2021). *Splinedist: Automated Cell Segmentation with Spline Curves*. Cold Spring Harbour Laboratories. doi:10.1101/2020.10.27.357640

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B. A., Karhohs, K. W., et al. (2018). Cellprofiler 3.0: Next-Generation Image Processing for Biology. *Plos Biol.* 16, e2005970–17. doi:10.1371/journal.pbio.2005970

Moen, E., Borba, E., Miller, G., Schwartz, M., Bannon, D., Koe, N., et al. (2019). *Accurate Cell Tracking and Lineage Construction in Live-Cell Imaging Experiments with Deep Learning*. Cold Spring Harbour Laboratories. doi:10.1101/803205

Mura, M., Feillet, C., Bertolusso, R., Delaunay, F., and Kimmel, M. (2019). Mathematical Modelling Reveals Unexpected Inheritance and Variability Patterns of Cell Cycle Parameters in Mammalian Cells. *Plos Comput. Biol.* 15, e1007054–26. doi:10.1371/journal.pcbi.1007054

Narayana, M., and Haverkamp, D. (2007). "A Bayesian Algorithm for Tracking Multiple Moving Objects in Outdoor Surveillance Video," in 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17-22 June 2007. doi:10.1109/cvpr.2007.383446

Norman, M., Wisniewska, K. A., Lawrenson, K., Garcia-Miranda, P., Tada, M., Kajita, M., et al. (2012). Loss of Scribble Causes Cell Competition in Mammalian Cells. *J. Cel Sci.* 125, 59–66. doi:10.1242/jcs.085803

Puliafito, A., Hufnagel, L., Neveu, P., Streichan, S., Sigal, A., Fygenson, D. K., et al. (2012). Collective and Single Cell Behavior in Epithelial Contact Inhibition. *Proc. Natl. Acad. Sci.* 109, 739–744. doi:10.1073/pnas.1007809109

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes Computer Sci. (including subseries Lecture Notes Artif. Intelligence Lecture Notes Bioinformatics)* 9351, 234–241. doi:10.1007/978-3-319-24574-4_28

Sandler, O., Mizrahi, S. P., Weiss, N., Agam, O., Simon, I., and Balaban, N. Q. (2015). Lineage Correlations of Single Cell Division Time as a Probe of Cell-Cycle Dynamics. *Nature* 519, 468–471. doi:10.1038/nature14318

Schiegg, M., Hanslovsky, P., Haubold, C., Koethe, U., Hufnagel, L., and Hamprecht, F. A. (2015). Graphical Model for Joint Segmentation and Tracking of Multiple Dividing Cells. *Bioinformatics* 31, 948–956. doi:10.1093/bioinformatics/btu764

Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. (2018). Cell Detection with star-convex Polygons. *Lecture Notes Computer Sci. (including subseries Lecture Notes Artif. Intelligence Lecture Notes Bioinformatics) 11071 LNCS* 11071, 265–273. doi:10.1007/978-3-030-00934-2_30

Skylaki, S., Hilsenbeck, O., and Schroeder, T. (2016). Challenges in Long-Term Imaging and Quantification of Single-Cell Dynamics. *Nat. Biotechnol.* 34, 1137–1144. doi:10.1038/nbt.3713

Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Winston, P., Bokota, G., et al. (2021). *Napari/Napari: 0.4.7rc1*. doi:10.5281/zenodo.4633896

Stegmaier, J., Amat, F., Lemon, W. C., McDole, K., Wan, Y., Teodoro, G., et al. (2016). Real-Time Three-Dimensional Cell Segmentation in Large-Scale Microscopy Data of Developing Embryos. *Developmental Cel* 36, 225–240. doi:10.1016/j.devcel.2015.12.028

Stringer, C., Wang, T., Michaelos, M., and Pachitariu, M. (2020). Cellpose: a Generalist Algorithm for Cellular Segmentation. *Nat. Methods* 18 (1), 100–106. doi:10.1101/2020.02.02.93123810.1038/s41592-020-01018-x

Sugawara, K., Cevrim, C., and Averof, M. (2021). *Tracking Cell Lineages in 3D by Incremental Deep Learning*. Cold Spring Harbour Laboratories. doi:10.1101/2021.02.26.432552

Tinevez, J.-Y., Perry, N., Schindelin, J., Hoopes, G. M., Reynolds, G. D., Laplantine, E., et al. (2017). TrackMate: An Open and Extensible Platform for Single-Particle Tracking. *Methods* 115, 80–90. doi:10.1016/j.ymeth.2016.09.016

Tinevez, J. Y. (2021). *Mastodon*. Available at https://github.com/mastodon-sc/mastodon.

Tsai, H.-F., Gajda, J., Sloan, T. F. W., Rares, A., and Shen, A. Q. (2019). Usiigaci: Instance-Aware Cell Tracking in Stain-free Phase Contrast Microscopy Enabled by Machine Learning. *SoftwareX* 9, 230–237. doi:10.1016/j.softx.2019.02.007

Ulman, V., Maška, M., Magnusson, K. E. G., Ronneberger, O., Haubold, C., Harder, N., et al. (2017). An Objective Comparison of Cell-Tracking Algorithms. *Nat. Methods* 14, 1141–1152. doi:10.1038/nmeth.4473

Wen, C., Miura, T., Fujie, Y., Teramoto, T., Ishihara, T., and Kimura, K. D. (2018). *Deep-learning-based Flexible Pipeline for Segmenting and Tracking Cells in 3D Image Time Series for Whole Brain Imaging*. Cold Spring Harbour Laboratories, 385567. doi:10.1101/385567

Wolff, C., Tinevez, J.-Y., Pietzsch, T., Stamataki, E., Harich, B., Guignard, L., et al. (2018). Multi-view Light-Sheet Imaging and Tracking with the MaMuT Software Reveals the Cell Lineage of a Direct Developing Arthropod Limb. *eLife* 7, 1–31. doi:10.7554/eLife.34410

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 Computer Science

ORIGINAL RESEARCH
published: 12 November 2021
doi: 10.3389/fcomp.2021.757653

Diffusion State Transitions in Single-Particle Trajectories of MET Receptor Tyrosine Kinase Measured in Live Cells

*Johanna V. Rahm[1], Sebastian Malkusch[2], Ulrike Endesfelder[3,4], Marina S. Dietz[1] and Mike Heilemann[1]**

[1]Institute of Physical and Theoretical Chemistry, Goethe University, Frankfurt, Germany, [2]Institute of Clinical Pharmacology, Goethe University, Frankfurt, Germany, [3]Department of Physics, Carnegie Mellon University, Pittsburgh, PA, United States, [4]Institute for Microbiology and Biotechnology, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany

Single-particle tracking enables the analysis of the dynamics of biomolecules in living cells with nanometer spatial and millisecond temporal resolution. This technique reports on the mobility of membrane proteins and is sensitive to the molecular state of a biomolecule and to interactions with other biomolecules. Trajectories describe the mobility of single particles over time and provide information such as the diffusion coefficient and diffusion state. Changes in particle dynamics within single trajectories lead to segmentation, which allows to extract information on transitions of functional states of a biomolecule. Here, mean-squared displacement analysis is developed to classify trajectory segments into immobile, confined diffusing, and freely diffusing states, and to extract the occurrence of transitions between these modes. We applied this analysis to single-particle tracking data of the membrane receptor MET in live cells and analyzed state transitions in single trajectories of the un-activated receptor and the receptor bound to the ligand internalin B. We found that internalin B-bound MET shows an enhancement of transitions from freely and confined diffusing states into the immobile state as compared to un-activated MET. Confined diffusion acts as an intermediate state between immobile and free, as this state is most likely to change the diffusion state in the following segment. This analysis can be readily applied to single-particle tracking data of other membrane receptors and intracellular proteins under various conditions and contribute to the understanding of molecular states and signaling pathways.

Keywords: single-particle tracking, single-trajectory analysis, membrane receptors, MET receptor, single-molecule imaging, diffusion states

OPEN ACCESS

Edited by:
Virginie Uhlmann, European Bioinformatics Institute (EMBL-EBI), United Kingdom

Reviewed by:
Jean-Yves Tinevez, Institut Pasteur, France
Daniel Sage, École Polytechnique Fédérale de Lausanne, Switzerland

****Correspondence:***
Mike Heilemann
heileman@chemie.uni-frankfurt.de

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *12 August 2021*
Accepted: *25 October 2021*
Published: *12 November 2021*

Citation:
Rahm JV, Malkusch S, Endesfelder U, Dietz MS and Heilemann M (2021) Diffusion State Transitions in Single-Particle Trajectories of MET Receptor Tyrosine Kinase Measured in Live Cells. Front. Comput. Sci. 3:757653. doi: 10.3389/fcomp.2021.757653

INTRODUCTION

Cells sense their environment through membrane proteins, and extracellular stimuli are translated into intracellular signaling cascades and a cellular response. This process often begins with ligands that bind to membrane receptors, induce receptor oligomerization, and recruit other proteins such as co-receptors. The formation of receptor oligomers and signaling platforms reduce the receptor mobility and change its diffusion behavior (Stone et al., 2017; Dietz and Heilemann, 2019). Single-particle tracking (SPT) is a method to measure and to reveal subtle changes in the diffusion of

membrane receptors in cells and at the molecular level (Manzo and Garcia-Parajo, 2015; Shen et al., 2017). SPT requires low molecular densities, in order to allow single-molecule detection and assignment of these into single-protein trajectories. Such low molecular densities can be achieved by substoichiometric labeling, by the introduction of a photoactivatable fluorophore, or by using transiently binding labels that specifically target the membrane protein (Manley et al., 2008; Giannone et al., 2010). SPT provides information on diffusion coefficients and on the type of motion, i.e. free diffusion, spatially confined movement, and immobile particles (Michalet, 2010). It may also occur that a molecule switches between different diffusion states within a single trajectory; such transitions can be analyzed by comparing the experimental dataset to Monte Carlo simulations (Wieser et al., 2008), using hidden Markov models (Persson et al., 2013; Sungkaworn et al., 2017; Liu et al., 2019), analytic diffusion distribution analysis (Vink et al., 2020), local MSD exponent values (Hubicka and Janczura, 2020), and unsupervised Gibbs sampling (Karslake et al., 2021).

Receptor tyrosine kinases (RTKs) constitute a family of membrane receptors comprising 58 different proteins (Lemmon and Schlessinger, 2010). One subfamily is the MET receptor family containing the hepatocyte growth factor receptor, also known as MET. MET was first discovered as an oncogene in 1984 (Cooper et al., 1984). The role of MET together with its physiological ligand hepatocyte growth factor/scatter factor (HGF/SF) is manifold: It is essential in embryogenesis, is involved in growth, and regulates cell migration (Bladt et al., 1995; Schmidt et al., 1995; Uehara et al., 1995). MET overexpression was found to be of relevance in several cancers and is targeted in cancer therapy (Ichimura et al., 1996; Goyal et al., 2013; Mo and Liu, 2017). Next to its canonical ligand HGF, MET is targeted by the surface protein internalin B (InlB) secreted by the pathogenic bacterium *Listeria monocytogenes* that causes human listeriosis (Braun et al., 1998). InlB triggers similar cellular responses as HGF/SF and induces bacterial invasion into hepatocytes (Dramsi et al., 1995; Shen et al., 2000; Niemann et al., 2007).

Here, we apply a segmentation analysis to single-molecule trajectories of un-activated and InlB-bound MET and extract the diffusion states and transitions between these diffusion states. To follow activated MET, we used the N-terminal internalin domain of InlB ($InlB_{321}$) that binds to the extracellular domain of MET and induces MET phosphorylation (Banerjee et al., 2004; Niemann et al., 2007; Ferraris et al., 2010; Dietz et al., 2013), and that we used in previous work to measure the diffusion of MET in living HeLa cells with single-particle tracking (Harwardt et al., 2017). We found that MET bound to InlB diffuses slower than resting receptors and that the immobile population increases. This immobilization was assigned to interactions with the actin cytoskeleton as well as to recruitment of MET to endocytosis sites. Using a segmentation approach, we now present an extended analysis of these data by taking into account that single receptors may switch between different diffusive states within single trajectories. For this analysis, single trajectories were divided into segments showing uniform movement. These segments were analyzed separately with regard to their diffusion mode (free, confined, immobile) (Rossier et al., 2012; Harwardt et al., 2017; Orré et al., 2021). In addition, we extracted the transitions between different segments within single trajectories, which report on functional transitions of the MET receptor signaling complex. For MET, we found that upon InlB activation the immobile state becomes more stable and transitions into immobile states occur more often. The confined diffusion state acts as an intermediate state between immobile and free, as this state is most likely to change the diffusion state in the following segment. This straight-forward analysis routine can be transferred to SPT data of other biological targets.

METHODS

Data Acquisition

The SPT data used within this study, together with experimental details on data acquisition and sample preparation, were previously published (Harwardt et al., 2017). In brief, the universal point accumulation for imaging in nanoscale topography (uPAINT) method (Giannone et al., 2010) was applied to measure the dynamics of the MET receptor in living HeLa cells. For the resting receptor, an ATTO 647N-labeled, non-activating Fab antibody fragment was used. The ligand-bound state was probed using the $InlB_{321}$ ligand site-specifically labeled with ATTO 647N which was fully functional (Dietz et al., 2013; Dietz et al., 2019). Imaging was performed in total internal reflection fluorescence (TIRF) mode using an N-STORM microscope (Nikon, Japan). For both un-activated MET and InlB-bound MET, 60 cells were analyzed.

Single-Molecule Localization

The MET receptor was targeted with fluorescent labels and its position in a cell membrane determined by analyzing image stacks with the *ThunderSTORM* plugin (version dev-2016-09-10-b1) (Ovesný et al., 2014) implemented in the image processing program *Fiji* (Schindelin et al., 2015). Camera settings were adjusted according to the manufacturer's manual and the base level was estimated by averaging the pixel intensity with the shutter closed. Deviations from *ThunderSTORM* default settings are the chosen fitting method "maximum likelihood", activated "multi-emitter fitting analysis" with a "maximum numbers of molecules per fitting region" of 3 and a "limit intensity range" spanning the 2-sigma interval of the photon distribution in log-space, extracted from detected emitters with "multi-emitter fitting analysis" disabled. The localizations were filtered by applying "remove duplicates".

Single-Particle Tracking

Trajectories of MET receptors were obtained by loading single-molecule localization data provided by *ThunderSTORM* into the *swift* tracking software (version 0.4.2) (Endesfelder et al., manuscript in prep). Parameters for *swift* analysis were determined using the *SPTAnalyser* software. A detailed

description is added to the manual at https://github.com/JohannaRahm/SPTAnalyser. The parameters "diffraction_limit" = 14 nm, "exp_displacement" = 85 nm (Fab)/75 nm (InlB), "p_bleach" = 0.010 (Fab)/0.014 (InlB), and "p_switch" = 0.01 were set globally for all cells. The parameters "exp_noise_rate" and "precision" were calculated individually per cell. *swift* divides trajectories into segments if the diffusion behavior of the particle changes.

Diffusion State Analysis

The diffusion state analysis was performed with *SPTAnalyser*. Diffusion coefficients of individual segments were calculated by optimizing the parameters of a linear diffusion model on the basis of the first four time steps of the mean squared displacement using the method of least squares (**Eq. 1**). Segments with a diffusion coefficient below 0 were discarded.

$$MSD(\Delta t) = 4D\Delta t \tag{1}$$

Segments with a minimum length of 20 frames (400 ms) were classified into diffusion states as previously reported (Rossier et al., 2012; Harwardt et al., 2017; Orré et al., 2021). First, the segments were separated into immobile and mobile diffusion applying a diffusion coefficient threshold D_{min}. The threshold is derived from the dynamic localization error (**Eq. 2**) which was calculated for each cell, with average values of MSD(0) and diffusion coefficient D (Savin and Doyle, 2005; Michalet, 2010). The third quartile was used to determine D_{min} (**Eq. 3**), where n is the number of time steps used in the linear model to extract the diffusion coefficient. All segments with a diffusion coefficient below D_{min} = 0.0028 μm^2/s were classified as immobile.

$$\sigma_{dyn} = \sqrt{\frac{\langle MSD(0)\rangle + \frac{4}{3}\langle D\rangle \cdot dt}{4}} \tag{2}$$

$$D_{min} = \frac{\sigma_{dyn}^2}{4 \cdot n \cdot dt} \tag{3}$$

Mobile segments were separated by fitting 60% of the MSD plot with **Eq. 4**, where r_c is the confined diffusion radius and τ is a time constant. Segments with τ smaller than half the time interval used to compute the MSD (120 ms) are classified as confined diffusion, and values higher than that as free diffusion.

$$MSD(\Delta t) = \frac{4}{3} r_c^2 \cdot \left(1 - e^{-\Delta t/\tau}\right) \tag{4}$$

Transition Counting

For each single trajectory in which at least two segments were identified, the transition of the diffusion state between the segments was determined. For the three diffusion states of immobile (i), confined (c), and freely diffusing (f) particles, nine different transitions are distinguished: i-i, i-c, i-f, c-i, c-c, c-f, f-i, f-c, f-f. Segments with a length of less than 20 frames, or with a negative diffusion coefficient, were not classified. Transitions between a classified and an unclassified segment were neglected. Unclassified segments that occurred between two classified segments, and that had a length of up to 19 frames, were masked, and the transition between the segment before and after the unclassified segment was considered in the analysis. This means that all segments that were shorter or equal than the mask length of 19 frames were removed from the trajectories, and that transitions between the preceding and the succeeding segment were counted. The mask value was synchronized to the minimum length a segment must exceed to be classified into a diffusion state. Transition counts were normalized per cell and summed to one to compare the occurrences of transition types. Transition counts were normalized per diffusion state so that the counts of transition types proceeding from the same diffusion state summed up to one to compare the occurrences of diffusion states in adjacent segments.

Statistical Analysis

Mean values are listed with respective standard errors of the mean (SEM). Mean values and SEMs were determined for each cell and globally averaged. Nonparametric tests were chosen as populations partly rejected the hypothesis of being normally distributed (tested with the Shapiro-Wilk test for normality, significance level = 0.05). Wilcoxon signed rank tests were used to validate the comparison of distributions within a treatment group (**Supplementary Tables S2, S5**). Mann-Whitney U tests were used to validate the comparison of distributions from two treatment groups (**Supplementary Tables S1, S3, S4, S6**). Levels of significance were classified as follows: $p > 0.05$ no significant difference (n.s.), $p < 0.05$ significant difference (*), $p < 0.01$ very significant difference (**), $p < 0.001$ highly significant difference (***). All tests were performed with SciPy (version 1.6.2) (Virtanen et al., 2020).

Simulations

To evaluate the error rate of the diffusion state classification, simulations of single-particle trajectories were performed. For this purpose, the software *ermine* (Estimate Reaction-rates by Markov-based Investigation of Nanoscopy Experiments) was used to create simulations of trajectories of freely diffusing particles. The probability distribution was defined by the expectation value of the mean squared displacement r within a timestep t (**Eq. 5**). The apparent mean squared displacement r was calculated based on the apparent diffusion coefficient D and the static error ε (Savin and Doyle, 2005) (**Eq. 6**). In order to match the simulation close to the experimental data, the following parameters were chosen: for D, the average diffusion coefficient of 0.12 μm^2/s of the mobile population (confined and free) was used, t corresponds to the camera integration time of 0.02 s, τ to the time lag between two consecutive frames of 0.02 s, and ε to the average localization error of 29 nm.

$$p\left(r|\langle r^2\rangle\right) = \frac{2r}{\langle r^2\rangle} \cdot e^{-\left(\frac{r^2}{\langle r^2\rangle}\right)} \tag{5}$$

$$\langle r^2\rangle = 4D\left(t - \frac{\tau}{3}\right) + 4\varepsilon^2 \tag{6}$$

The error rate of the diffusion state classification model was estimated by classifying the simulated trajectories of freely

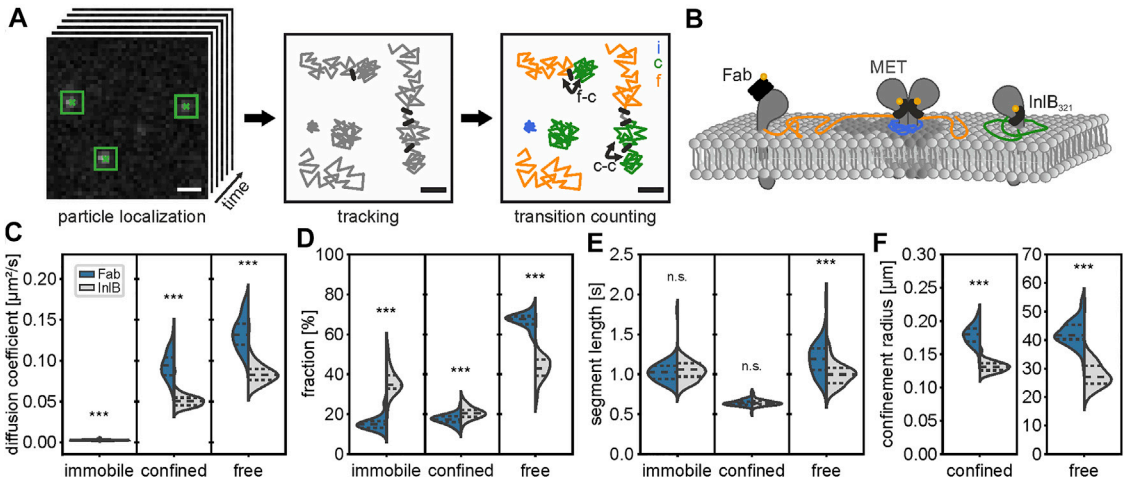


FIGURE 1 | Analysis pipeline of single-particle tracking data of the MET membrane receptor. **(A)** Diffraction limited signal of membrane-bound fluorescently labeled receptors is localized with subpixel accuracy (scale bar 1 µm) (left). Localizations are connected over time to trajectories and changes in diffusion behavior within a trajectory give separated segments (schematic scale bar 100 nm) (middle). Segments are classified into the diffusion states immobile (blue), confined (green), and free (orange) (right). Gray segments are below the necessary length to be classified and have no diffusion state label. Transitions between diffusion states are counted. **(B)** Membrane bound MET receptors are labeled extracellularly with a monoclonal Fab fragment that binds but does not activate and with InlB that binds and activates the receptor. Both ligands carry ATTO 647N as a fluorescent label. The movement of bound labels is detected and analyzed. The segment properties **(C)** diffusion coefficient, **(D)** population of diffusion states, **(E)** segment length, and **(F)** confinement radius are displayed as violin plots; dashed lines mark the quartiles. For each condition, Fab (blue) and InlB (gray), the average segment values of 60 cells are displayed.

diffusing particles into confined or free by fitting their MSD values with **Eq. 4** (see *Diffusion State Analysis* Section). The false negative rate was calculated as the number of confined classified trajectories divided by the sum of confined and free classified trajectories.

Availability

The analysis procedure introduced in this work can be straightforward applied to other single-particle tracking data. Localizations can be detected with *rapidSTORM* (Wolter et al., 2012) or *ThunderSTORM* (Ovesný et al., 2014) and connected to trajectories with *swift*. *Swift* version 0.4.2, used in this manuscript, and all subsequent versions of the *swift* software, as well as documentation and test data sets, can be obtained on the swift beta-testing repository (http://bit.ly/swifttracking). The home-written software *SPTAnalyser* in Python (3.7.6) estimates parameters for tracking with *swift* and executes diffusion state analysis and transition counting. *SPTAnalyser* has a graphical user interface with adaptable analysis parameters and assists in processing large amounts of data by creating macros for *ThunderSTORM* and batch files for *swift*. *SPTAnalyser* is compatible with *PALMTracer* (Bordeaux Imaging Center), which is a software for localization and tracking available as a plugin for MetaMorph (Molecular Devices, Sunnyvale, CA, USA). The source code of *SPTAnalyser*, together with a detailed manual, is available from https://github.com/JohannaRahm/SPTAnalyser. Simulations of trajectories of freely diffusing particles were conducted with *ermine* version 0.1 (https://github.com/SMLMS/ermine-tutorial).

RESULTS

Extraction of Different Diffusion States Within Single-Molecule Trajectories

We developed a data analysis workflow that extracts transitions between diffusion states from single-particle trajectories (**Figure 1A**, **Supplementary Figure S1**). We applied this analysis to single-particle tracking data of MET receptors in live HeLa cells recorded using the uPAINT principle (Giannone et al., 2010). For that purpose, MET receptors were either labeled with a monoclonal Fab fragment, which binds to but does not activate the receptor, or with the bacterial ligand InlB, which binds and activates the receptor (**Figure 1B**). Both ligands were conjugated to the fluorophore ATTO 647N. The positions of the fluorophore labels were measured in live cells using TIRF microscopy and subsequently linked to trajectories (**Figure 1A**). Individual trajectories were divided into segments that exhibited uniform motion. Segments were classified as immobile (i), confined (c), and freely diffusing (f) states and transitions between diffusion states within single trajectories were analyzed.

Segments of single trajectories of Fab- and InlB-bound receptors exhibit different properties in terms of their mobility, population of diffusion states, lengths, and confinement radii (**Figures 1C–F**). Diffusion states (free, confined, immobile) were determined by analyzing the MSD plots of the segments (for details see Methods). The diffusion coefficients of the InlB/MET complexes are significantly smaller compared to Fab/MET for the confined (D_{InlB} = 0.051 ± 0.003 µm^2/s vs D_{Fab} = 0.094 ± 0.007 µm^2/s) and free (D_{InlB} =

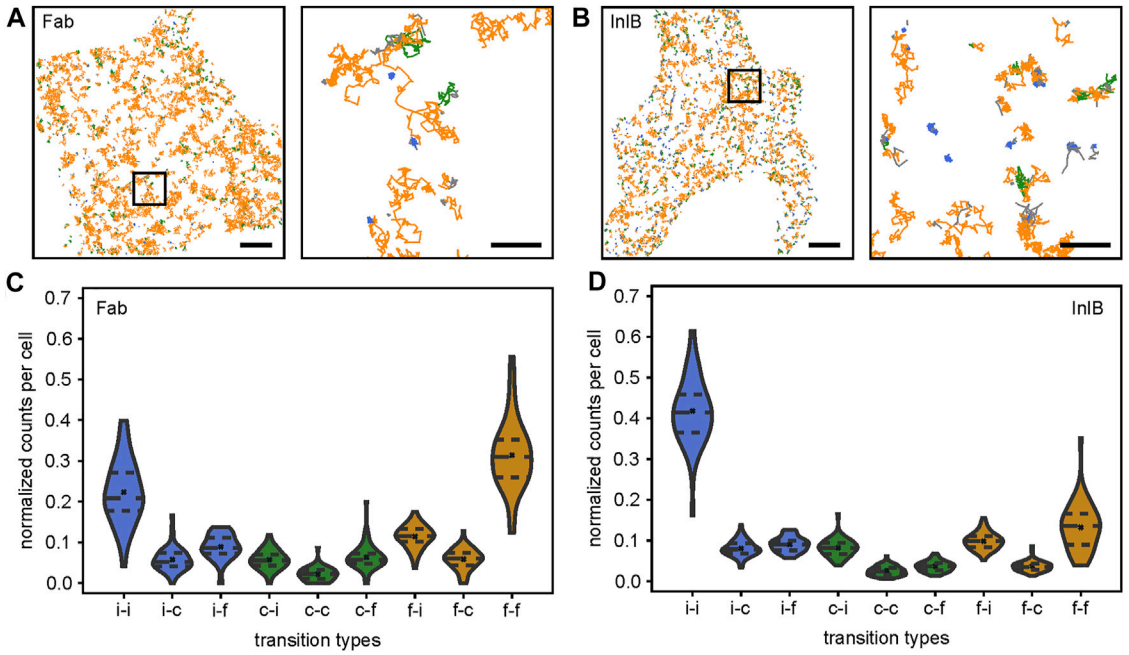


FIGURE 2 | Single-molecule trajectories and quantification of transitions within trajectories. Single-molecule trajectories of **(A)** Fab-bound and **(B)** InlB-bound MET receptors within exemplary cells. Diffusion states of segments are highlighted in colors (free: orange, confined: green, immobile: blue). Highlighted regions in the overview images are shown as zoom-ins (right). In the zoom-ins only trajectories showing at least one transition are displayed. Scale bars 5 μm, zoom-ins 1 μm. Violin plots of the nine different diffusion state transitions between segments within trajectories in **(C)** Fab-bound and **(D)** InlB-bound cells (i = immobile, c = confined, f = free). Counts are normalized to one per cell. For each condition 60 cells were analyzed. Dashed lines mark the quartiles, crosses correspond to the means.

0.084 ± 0.003 μm^2/s vs D_{Fab} = 0.134 ± 0.004 μm^2/s) population (**Figure 1C**). The diffusion coefficients for the immobile populations are smaller than the precision of the method and result from segments below the detection limit of mobility (see *Methods*). Upon activation with InlB, the population of the freely diffusing particles is reduced and driven towards the immobile state (**Figure 1D**). Segment lengths are drastically shorter for confined segments compared to the other two diffusion states (**Figure 1E**). For example, in InlB-treated cells, a segment classified as confined diffusion lasts an average of 0.64 ± 0.02 s compared to 1.19 ± 0.03 s for immobile and 1.00 ± 0.04 s for a free diffusion. InlB-bound receptors generally move in a more confined manner, as confinement radii calculated for the confined and free populations are smaller compared to un-activated MET (**Figure 1F**). Interestingly, the confinement radii of the free diffusion state are in the order of magnitude of the cell sizes.

To evaluate the accuracy of the classification model, simulated trajectories of freely diffusing particles were classified. An error rate was calculated from freely diffusing particles particles that were classified as confined diffusion (**Supplementary Figure S2**). This error rate decreased with increasing trajectory length. In addition, trajectories of freely diffusing particles were simulated with number and trajectory length corresponding to the distribution of trajectories in the Fab experiments, resulting in an error rate of 15%. This suggests the possibility that trajectories of freely diffusing particles contribute to the confined population and reduce the average trajectory length, as a misclassification is more likely with shorter trajectories.

The number of transition events observed for all trajectories of a cell is 184 ± 11 within a measurement period of 20 s. Compared to the average number of 1440 ± 90 trajectories per cell, this number appears relatively small, which is because such events rarely occur within the observed time window of a trajectory (1.36 ± 0.06 s). More transition events occur in longer trajectories (**Supplementary Figure S1A**). However, 70% of the trajectories do not change their diffusion mode and consist of only one segment (**Supplementary Figure S1B**). The number of transition counts increases by up to 30% by masking unclassified segments, i.e. segments with a length below the threshold of 20 frames. For this, the transitions of adjacent segments with a defined diffusion state around the masked segment are counted. Without masking, the average is 22.2 ± 0.6% of transitions between segments with a defined diffusion state. With masking, this value is increased to 71.6 ± 0.7% (**Supplementary Figure S1C**). Mostly no significant changes are observed between the relative frequencies of transitions when masking or without masking (**Supplementary Figures S1D,E**). Only transitions between immobile and free diffusion benefited slightly less from masking than those of the other transition types.

MET Receptor Activation With InlB Changes Diffusion State Transitions Between Segments

Segments of single-molecule trajectories of Fab-bound as well as InlB-bound MET receptors were classified into freely diffusing, confined moving, and immobile particles as described above. Exemplary cells with color-coded segments are shown in **Figures 2A,B** for resting and activated MET, respectively. In the zoom-ins only trajectories with at least one transition are displayed. In InlB-treated cells, the number of confined and especially immobile segments increases in comparison to Fab-treated cells, while at the same time the occurrence of freely diffusing particles is significantly lower. In addition, an increased confinement of InlB-bound receptors is visible. These observations are in accordance with the increased fractions of immobile and confined receptors (**Figure 1D**) and the decreased confinement radius of InlB-bound MET trajectories.

In the next step, we quantified the probability of specific transitions between segments in individual trajectories (**Figures 2C,D**). From this analysis, we found a probability of 22 ± 1% for Fab-bound receptors and 42 ± 2% for InlB-bound receptors that an immobile particle stays immobile. A similar observation is made for freely diffusing receptors, which mainly stay in this diffusion state (Fab: 31 ± 2%, InlB: 13.2 ± 0.8%). Homogeneous transitions of the confined diffusion state are less probable (Fab: 2.2 ± 0.3%, InlB: 2.6 ± 0.2%). This is interesting, as the confined and immobile states for Fab-bound receptors (immobile: 15.6 ± 0.5%, confined: 17.4 ± 0.4%) are nearly equally populated (**Figure 1D**), but significantly less homogeneous transitions occur in the confined population. Regarding heterogeneous transitions, transitions from free to immobile (Fab: 11.4 ± 0.4%, InlB: 9.8 ± 0.3%), immobile to free (Fab: 8.9 ± 0.4%, InlB: 9.1 ± 0.3%), and confined to free for Fab-bound MET (6.3 ± 0.4%) and confined to immobile for InlB-bound MET (8.2 ± 0.3%) are most frequent.

Interestingly, when comparing the frequencies of transitions between resting and InlB-bound MET receptors they mostly differ highly significantly (**Supplementary Figure S2**). Only the transitions from immobile to free and from confined to confined do not change significantly. The transition from immobile to immobile segments, from immobile to confined segments, as well as from confined to immobile segments increases for the activated cells. At the same time, transitions to the freely diffusing state occur less probable out of the InlB-activated state.

To visualize the differences between the different diffusion states, we normalized the transitions with regard to the respective diffusion state (**Supplementary Figure S3**). For both, Fab-bound and InlB-bound MET receptor trajectories, it clearly shows that the immobile and the free state are relatively stable diffusion states, which we infer from the high probability that an immobile particle stays immobile in the next segment and a freely diffusing molecule remains freely diffusing (**Supplementary Figures S3 A,B**). The confined state appears to be a more intermediate state: a confined diffusing receptor very likely changes its diffusion state in the next segment, either getting immobilized or switching to free diffusion. When comparing resting and InlB-activated MET mobility most transitions significantly change (**Supplementary Figure 3C**). Transitions towards the immobile state become more likely and to the freely diffusing state less probable in activated cells. The transitions involving the confined diffusing state change less significantly.

DISCUSSION

We report an analysis method for single-particle tracking data that resolves segments of different diffusional states within single trajectories. The method is sensitive to report segments of free, confined, and immobile states within single trajectories, and transitions between these diffusion states. This allowed us to relate dynamic information on protein mobility to functional states of a protein in a membrane, e.g. the immobilization upon binding of a ligand to a receptor. This additional information from single-particle tracking data complements the available portfolio on analyzing mobility data of single proteins (Rossier et al., 2012; Calebiro et al., 2013; Ibach et al., 2015; Sungkaworn et al., 2017).

As a showcase example, we investigated the diffusion of the MET receptor in living HeLa cells by analyzing available single-particle tracking data of resting and InlB-activated MET (Harwardt et al., 2017). Our analysis reports similar diffusion coefficients for resting and InlB-bound MET. In addition, we were able to segment trajectories and to reveal transitions between diffusion states within single trajectories; this information was so far averaged out by a global MSD analysis of single-particle trajectories. The analysis of segments in single MET receptor trajectories revealed that upon activation, MET transits from a free diffusion state to confined and immobile states and the immobile state becomes more stable, which is in line with the canonical model of receptor tyrosine kinase activation and internalization (Li et al., 2005; Chung et al., 2010). Interestingly, we found that the confined state has a short lifetime which is reflected in the segment lengths as well as the transition probabilities. This diffusion state can be seen as an intermediate state of MET. Upon entry into this state, it is probable that the receptor is soon either immobilized, e.g. prior to endocytosis, or returns to a highly mobile state, e.g. searching for interaction partners.

Our analysis procedure can be applied to single-particle tracking data of other molecules and provides straight-forward access to transitions in the mobility of proteins that can be related to functional states. Future developments may focus on extending the trajectory length and extracting the kinetics of transitions within single trajectories. This could be achieved by either using more stable fluorescent probes such as quantum dots (Hagen et al., 2009; Li et al., 2012; Cognet et al., 2014), or by recording single-molecule data with very low illumination intensity and in combination with image analysis-assisted

localization through e.g. denoising algorithms (Kefer et al., 2021). Another exciting extension is dual-color SPT (Wilmes et al., 2015, 2020), which in combination with segmentation analysis may relate changes in diffusion states to molecular interactions such as the formation of transient complexes between two receptors.

DATA AVAILABILITY STATEMENT

The dataset analyzed in this study can be found at the EMBL/EBI BioStudies server; https://www.ebi.ac.uk/biostudies/studies/S-BSST712.

AUTHOR CONTRIBUTIONS

MH conceptualized the study. JR, MD, and MH conceived the experiments. UE provided software, and JR, UE, and SM developed data analysis protocols. JR performed data analysis. All authors discussed the results, contributed to manuscript revision, read, and approved the final submitted version.

FUNDING

MH and JR acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project number 414985841, GRK 2566.

ACKNOWLEDGMENTS

We thank Marie-Lena Harwardt for providing the single-particle tracking data for the analysis and discussing their analysis, Claudia Catapano for testing the SPTAnalyser software and valuable input, as well as Marc Endesfelder and Bartosz Turkowyd for helpful discussions on the tracking analysis.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2021.757653/full#supplementary-material

REFERENCES

Banerjee, M., Copp, J., Vuga, D., Marino, M., Chapman, T., van der Geer, P., et al. (2004). GW Domains of the Listeria Monocytogenes Invasion Protein InlB Are Required for Potentiation of Met Activation. *Mol. Microbiol.* 52, 257–271. doi:10.1111/j.1365-2958.2003.03968.x

Bladt, F., Riethmacher, D., Isenmann, S., Aguzzi, A., and Birchmeier, C. (1995). Essential Role for the C-Met Receptor in the Migration of Myogenic Precursor Cells into the Limb Bud. *Nature* 376, 768–771. doi:10.1038/376768a0

Braun, L., Ohayon, H., and Cossart, P. (1998). The InlB Protein ofListeria Monocytogenesis Sufficient to Promote Entry into Mammalian Cells. *Mol. Microbiol.* 27, 1077–1087. doi:10.1046/j.1365-2958.1998.00750.x

Calebiro, D., Rieken, F., Wagner, J., Sungkaworn, T., Zabel, U., Borzi, A., et al. (2013). Single-molecule Analysis of Fluorescently Labeled G-Protein-Coupled Receptors Reveals Complexes with Distinct Dynamics and Organization. *Proc. Natl. Acad. Sci.* 110, 743–748. doi:10.1073/pnas.1205798110

Chung, I., Akita, R., Vandlen, R., Toomre, D., Schlessinger, J., and Mellman, I. (2010). Spatial Control of EGF Receptor Activation by Reversible Dimerization on Living Cells. *Nature* 464, 783–787. doi:10.1038/nature08827

Cognet, L., Leduc, C., and Lounis, B. (2014). Advances in Live-Cell Single-Particle Tracking and Dynamic Super-resolution Imaging. *Curr. Opin. Chem. Biol.* 20, 78–85. doi:10.1016/j.cbpa.2014.04.015

Cooper, C. S., Park, M., Blair, D. G., Tainsky, M. A., Huebner, K., Croce, C. M., et al. (1984). Molecular Cloning of a New Transforming Gene from a Chemically Transformed Human Cell Line. *Nature* 311, 29–33. doi:10.1038/311029a0

Dietz, M. S., Haße, D., Ferraris, D. M., Göhler, A., Niemann, H. H., and Heilemann, M. (2013). Single-molecule Photobleaching Reveals Increased MET Receptor Dimerization upon Ligand Binding in Intact Cells. *BMC Biophys.* 6, 6. doi:10.1186/2046-1682-6-6

Dietz, M. S., and Heilemann, M. (2019). Optical Super-resolution Microscopy Unravels the Molecular Composition of Functional Protein Complexes. *Nanoscale* 11, 17981–17991. doi:10.1039/c9nr06364a

Dietz, M. S., Wehrheim, S. S., Harwardt, M.-L. I. E., Niemann, H. H., and Heilemann, M. (2019). Competitive Binding Study Revealing the Influence of Fluorophore Labels on Biomolecular Interactions. *Nano Lett.* 19, 8245–8249. doi:10.1021/acs.nanolett.9b03736

Dramsi, S., Biswas, I., Maguin, E., Braun, L., Mastroeni, P., and Cossart, P. (1995). Entry of Listeria Monocytogenes into Hepatocytes Requires Expression of InlB, a Surface Protein of the Internalin Multigene Family. *Mol. Microbiol.* 16, 251–261. doi:10.1111/j.1365-2958.1995.tb02297.x

Ferraris, D. M., Gherardi, E., Di, Y., Heinz, D. W., and Niemann, H. H. (2010). Ligand-mediated Dimerization of the Met Receptor Tyrosine Kinase by the Bacterial Invasion Protein InlB. *J. Mol. Biol.* 395, 522–532. doi:10.1016/j.jmb.2009.10.074

Giannone, G., Hosy, E., Levet, F., Constals, A., Schulze, K., Sobolevsky, A. I., et al. (2010). Dynamic Superresolution Imaging of Endogenous Proteins on Living Cells at Ultra-high Density. *Biophysical J.* 99, 1303–1310. doi:10.1016/j.bpj.2010.06.005

Goyal, L., Muzumdar, M. D., and Zhu, A. X. (2013). Targeting the HGF/c-MET Pathway in Hepatocellular Carcinoma. *Clin. Cancer Res.* 19, 2310–2318. doi:10.1158/1078-0432.ccr-12-2791

Hagen, G. M., Lidke, K. A., Rieger, B., Lidke, D. S., Caarls, W., Arndt-Jovin, D. J., et al. (2009). Dynamics of Membrane Receptors: Single-Molecule Tracking of Quantum Dot Liganded Epidermal Growth Factor. *Single Molecule Dyn. Life Sci.* 117, 117–130. doi:10.1002/9783527626137.ch6

Harwardt, M.-L. I. E., Young, P., Bleymüller, W. M., Meyer, T., Karathanasis, C., Niemann, H. H., et al. (2017). Membrane Dynamics of Resting and Internalin B-Bound MET Receptor Tyrosine Kinase Studied by Single-Molecule Tracking. *FEBS Open Bio* 7, 1422–1440. doi:10.1002/2211-5463.12285

Hubicka, K., and Janczura, J. (2020). Time-dependent Classification of Protein Diffusion Types: A Statistical Detection of Mean-Squared-Displacement Exponent Transitions. *Phys. Rev. E* 101, 022107. doi:10.1103/physreve.101.022107

Ibach, J., Radon, Y., Gelléri, M., Sonntag, M. H., Brunsveld, L., Bastiaens, P. I. H., et al. (2015). Single Particle Tracking Reveals that EGFR Signaling Activity Is Amplified in Clathrin-Coated Pits. *PLoS One* 10, e0143162. doi:10.1371/journal.pone.0143162

Ichimura, E., Maeshima, A., Nakajima, T., and Nakamura, T. (1996). Expression Ofc-Met/HGF Receptor in Human Non-small Cell Lung Carcinomasin Vitroandin Vivoand its Prognostic Significance. *Jpn. J. Cancer Res.* 87, 1063–1069. doi:10.1111/j.1349-7006.1996.tb03111.x

Karslake, J. D., Donarski, E. D., Shelby, S. A., Demey, L. M., DiRita, V. J., Veatch, S. L., et al. (2021). SMAUG: Analyzing Single-Molecule Tracks with Nonparametric Bayesian Statistics. *Methods* 193, 16–26. doi:10.1016/j.ymeth.2020.03.008

Kefer, P., Iqbal, F., Locatelli, M., Lawrimore, J., Zhang, M., Bloom, K., et al. (2021). Performance of Deep Learning Restoration Methods for the Extraction of Particle Dynamics in Noisy Microscopy Image Sequences. *Mol. Biol. Cel* 32, 903–914. doi:10.1091/mbc.e20-11-0689

Lemmon, M. A., and Schlessinger, J. (2010). Cell Signaling by Receptor Tyrosine Kinases. *Cell* 141, 1117–1134. doi:10.1016/j.cell.2010.06.011

Li, H., Duan, Z.-W., Xie, P., Liu, Y.-R., Wang, W.-C., Dou, S.-X., et al. (2012). Effects of Paclitaxel on EGFR Endocytic Trafficking Revealed Using Quantum Dot Tracking in Single Cells. *PLoS One* 7, e45465. doi:10.1371/journal.pone.0045465

Li, N., Xiang, G.-S., Dokainish, H., Ireton, K., and Elferink, L. A. (2005). The Listeria Protein Internalin B Mimics Hepatocyte Growth Factor-Induced Receptor Trafficking. *Traffic* 6, 459–473. doi:10.1111/j.1600-0854.2005.00290.x

Liu, Y.-L., Chou, C.-K., Kim, M., Vasisht, R., Kuo, Y.-A., Ang, P., et al. (2019). Assessing Metastatic Potential of Breast Cancer Cells Based on EGFR Dynamics. *Sci. Rep.* 9, 3395. doi:10.1038/s41598-018-37625-0

Manley, S., Gillette, J. M., Patterson, G. H., Shroff, H., Hess, H. F., Betzig, E., et al. (2008). High-density Mapping of Single-Molecule Trajectories with Photoactivated Localization Microscopy. *Nat. Methods* 5, 155–157. doi:10.1038/nmeth.1176

Manzo, C., and Garcia-Parajo, M. F. (2015). A Review of Progress in Single Particle Tracking: from Methods to Biophysical Insights. *Rep. Prog. Phys.* 78, 124601. doi:10.1088/0034-4885/78/12/124601

Michalet, X. (2010). Mean Square Displacement Analysis of Single-Particle Trajectories with Localization Error: Brownian Motion in an Isotropic Medium. *Phys. Rev. E* 82, 041914. doi:10.1103/physreve.82.041914

Mo, H.-N., and Liu, P. (2017). Targeting MET in Cancer Therapy. *Chronic Dis. Translational Med.* 3, 148–153. doi:10.1016/j.cdtm.2017.06.002

Niemann, H. H., Jäger, V., Butler, P. J. G., van den Heuvel, J., Schmidt, S., Ferraris, D., et al. (2007). Structure of the Human Receptor Tyrosine Kinase Met in Complex with the Listeria Invasion Protein InlB. *Cell* 130, 235–246. doi:10.1016/j.cell.2007.05.037

Orré, T., Joly, A., Karatas, Z., Kastberger, B., Cabriel, C., Böttcher, R. T., et al. (2021). Molecular Motion and Tridimensional Nanoscale Localization of Kindlin Control Integrin Activation in Focal Adhesions. *Nat. Commun.* 12, 3104. doi:10.1038/s41467-021-23372-w

Ovesný, M., Křížek, P., Borkovec, J., Švindrych, Z., and Hagen, G. M. (2014). ThunderSTORM: a Comprehensive ImageJ Plug-In for PALM and STORM Data Analysis and Super-resolution Imaging. *Bioinformatics* 30, 2389–2390. doi:10.1093/bioinformatics/btu202

Persson, F., Lindén, M., Unoson, C., and Elf, J. (2013). Extracting Intracellular Diffusive States and Transition Rates from Single-Molecule Tracking Data. *Nat. Methods* 10, 265–269. doi:10.1038/nmeth.2367

Rossier, O., Octeau, V., Sibarita, J.-B., Leduc, C., Tessier, B., Nair, D., et al. (2012). Integrins β1 and β3 Exhibit Distinct Dynamic Nanoscale Organizations inside Focal Adhesions. *Nat. Cel Biol.* 14, 1057–1067. doi:10.1038/ncb2588

Savin, T., and Doyle, P. S. (2005). Static and Dynamic Errors in Particle Tracking Microrheology. *Biophysical J.* 88, 623–638. doi:10.1529/biophysj.104.042457

Schindelin, J., Rueden, C. T., Hiner, M. C., and Eliceiri, K. W. (2015). The ImageJ Ecosystem: An Open Platform for Biomedical Image Analysis. *Mol. Reprod. Dev.* 82, 518–529. doi:10.1002/mrd.22489

Schmidt, C., Bladt, F., Goedecke, S., Brinkmann, V., Zschiesche, W., Sharpe, M., et al. (1995). Scatter Factor/hepatocyte Growth Factor Is Essential for Liver Development. *Nature* 373, 699–702. doi:10.1038/373699a0

Shen, H., Tauzin, L. J., Baiyasi, R., Wang, W., Moringo, N., Shuang, B., et al. (2017). Single Particle Tracking: From Theory to Biophysical Applications. *Chem. Rev.* 117, 7331–7376. doi:10.1021/acs.chemrev.6b00815

Shen, Y., Naujokas, M., Park, M., and Ireton, K. (2000). InlB-Dependent Internalization of Listeria Is Mediated by the Met Receptor Tyrosine Kinase. *Cell* 103, 501–510. doi:10.1016/s0092-8674(00)00141-0

Stone, M. B., Shelby, S. A., and Veatch, S. L. (2017). Super-Resolution Microscopy: Shedding Light on the Cellular Plasma Membrane. *Chem. Rev.* 117, 7457–7477. doi:10.1021/acs.chemrev.6b00716

Sungkaworn, T., Jobin, M.-L., Burnecki, K., Weron, A., Lohse, M. J., and Calebiro, D. (2017). Single-molecule Imaging Reveals Receptor-G Protein Interactions at Cell Surface Hot Spots. *Nature* 550, 543–547. doi:10.1038/nature24264

Uehara, Y., Minowa, O., Mori, C., Shiota, K., Kuno, J., Noda, T., et al. (1995). Placental Defect and Embryonic Lethality in Mice Lacking Hepatocyte Growth Factor/scatter Factor. *Nature* 373, 702–705. doi:10.1038/373702a0

Vink, J. N. A., Brouns, S. J. J., and Hohlbein, J. (2020). Extracting Transition Rates in Particle Tracking Using Analytical Diffusion Distribution Analysis. *Biophysical J.* 119, 1970–1983. doi:10.1016/j.bpj.2020.09.033

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

Wieser, S., Axmann, M., and Schütz, G. J. (2008). Versatile Analysis of Single-Molecule Tracking Data by Comprehensive Testing against Monte Carlo Simulations. *Biophysical J.* 95, 5988–6001. doi:10.1529/biophysj.108.141655

Wilmes, S., Beutel, O., Li, Z., Francois-Newton, V., Richter, C. P., Janning, D., et al. (2015). Receptor Dimerization Dynamics as a Regulatory Valve for Plasticity of Type I Interferon Signaling. *J. Cel Biol.* 209, 579–593. doi:10.1083/jcb.201412049

Wilmes, S., Hafer, M., Vuorio, J., Tucker, J. A., Winkelmann, H., Löchte, S., et al. (2020). Mechanism of Homodimeric Cytokine Receptor Activation and Dysregulation by Oncogenic Mutations. *Science* 367, 643–652. doi:10.1126/science.aaw3242

Wolter, S., Löschberger, A., Holm, T., Aufmkolk, S., Dabauvalle, M.-C., van de Linde, S., et al. (2012). rapidSTORM: Accurate, Fast Open-Source Software for Localization Microscopy. *Nat. Methods* 9, 1040–1041. doi:10.1038/nmeth.2224

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 Computer Science

METHODS
published: 19 November 2021
doi: 10.3389/fcomp.2021.774396

Image Processing Filters for Grids of Cells Analogous to Filters Processing Grids of Pixels

Robert Haase [1,2]*

[1]Bio-image Analysis Technology Development Group, DFG Cluster of Excellence "Physics of Life", Technische Universität Dresden, Dresden, Germany, [2]Center for Systems Biology Dresden, Dresden, Germany

Intra- and extra-cellular processes shape tissues together. For understanding how neighborhood relationships between cells play a role in this process, having image processing filters based on these relationships would be beneficial. Those operations are known and their application to microscopy image data typically requires programming skills. User-friendly general purpose tools for pursuing image processing on a level of neighboring cells were yet missing. In this manuscript I demonstrate image processing filters which process grids of cells on tissue level and the analogy to their better known counter parts processing grids of pixels. The tools are available as part of free and open source software in the ImageJ/Fiji and napari ecosystems and their application does not require any programming experience.

Keywords: image processing, cells, tissues, region adjacency graphs, spatial analysis, imagej, Fiji, napari

OPEN ACCESS

Edited by:
Virginie Uhlmann,
European Bioinformatics Institute (EMBL-EBI), United Kingdom

Reviewed by:
Kyle Harrington,
Max Delbrück Center for Molecular Medicine, Germany
Sebastien Herbert,
University of Basel, Switzerland

***Correspondence:**
Robert Haase
robert.haase@tu-dresden.de

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *11 September 2021*
Accepted: *26 October 2021*
Published: *19 November 2021*

Citation:
Haase R (2021) Image Processing Filters for Grids of Cells Analogous to Filters Processing Grids of Pixels. Front. Comput. Sci. 3:774396. doi: 10.3389/fcomp.2021.774396

1 INTRODUCTION

Forces between intra-cellular structures, groups of cells and external structures are involved in forming an organism (Stern, 2004; Rauzi et al., 2008; Benton et al., 2013; Münster et al., 2019; Bhide et al., 2020). Processing images using filters for segmenting and analysing sub-cellular structures and cells is a common task for bio-image analysts. Furthermore, the concept of grouping pixels into superpixels for further analysis is well known in computer vision (Ren and Malik, 2003). Thus, it is at hand to process groups of pixels which represent cells and thus, sub-entities of images. Quantitative measurements derived from those digital representations of individual cells can deliver insights about a cells' identity. However, such measurements may not fully capture important aspects of tissue formation because of their spatially constrained nature. A cells' identity might depend on properties of neighboring cells (Toth et al., 2018). To unravel physical principles on tissue level, it is worthwhile to formulate image processing filters taking numeric parameters, such as quantitative measurements, of cells and cell-neighbors into account analogously to classical image processing filters taking the intensity of pixels and their neighbors as input.

Most image processing software packages are limited to process imaging data on pixel level using filters. Filters are defined as numerical operations which consume an image to produce another image. Images can be seen as grids of rectangular shaped picture elements, pixels, in 2D or volume elements, voxels, in 3D. Typically, filters take the neighboring pixels of every pixel into account to produce a new pixel value. For example, the mean average filter computes the average intensity of pixels around every pixel and saves the resulting value for each pixel in the new image. In common image processing software, various neighborhood specifications are used: Pixels touching a given reference pixel in corners, edges and voxels touching the given voxel in a plane can be seen as directly touching neighbors. Furthermore, pixels/voxels within a

rectangular/cuboidal or ellipsoidal shaped region with given radii around the reference pixel form the proximal neighborhood. Some image processing libraries also offer customizing neighborhood definitions to specific needs. Such a definition is then referred to as structuring element or footprint of a filter.

Combining pixels to objects is common practice and quantification of parameters of objects, such as size and shape, is straightforward. With that, we can visualize parameters of objects as intensity in images, so called parametric images. For example, a parametric image can show large objects with a higher intensity and small objects with a lower intensity. Objects can be cells and thus, studying quantitative parameters of cells in parametric images is straightforward. Studying relationships between cells in such parametric images is more complex and requires some form of so called region adjacency graph (RAG), an abstract representation of neighborhood relationships between cells. For example, measuring the mean area of cells in a given neighborhood around each cell requires determining the RAG of all cells in the tissue and then averaging the area of neighboring cells for each cell. If this would be straightforward, one could think of selecting all cells in a tissue which for example are smaller than the average area minus two times the standard deviation of the area of cells in a given neighborhood. In a more advanced setting, cell classification algorithms could take properties of cells into account that are within a given radius around the cell of interest. Image analysis workflows could benefit from incorporating such operations on tissue level and thus, might enable new perspectives when studying for example tissue development.

Software libraries in R and Python for processing RAGs, such as SPIAT (Yang et al., 2020) and NetworkX (Hagberg et al., 2008), are accessible to an audience with programming experience and offer powerful approaches to study spatial relationships between cells. Widely used open source tools with graphical user interfaces for processing quantitative measurements of cells and their neighbors such as CytoMap (Stoltzfus et al., 2020) typically operate on tabular data and are not integrated in image processing software. Another software package for digital pathology, QuPath (Bankhead et al., 2017), supports similar tools and is limited to two dimensional image data. Thus, the even though operations on RAGs in the cell/tissue context are well known, their application as general purpose tools for processing information between three dimensional cells were yet missing. In this article I present filters for processing objects such as cells, their neighbors and quantitative information derived from them targeting general-purpose use-cases. These filters can process two- and three-dimensional image data and are available for users without programming experience in the ImageJ/Fiji (Schneider et al., 2012; Schindelin et al., 2015) and napari (Sofroniew et al., 2020) ecosystems.

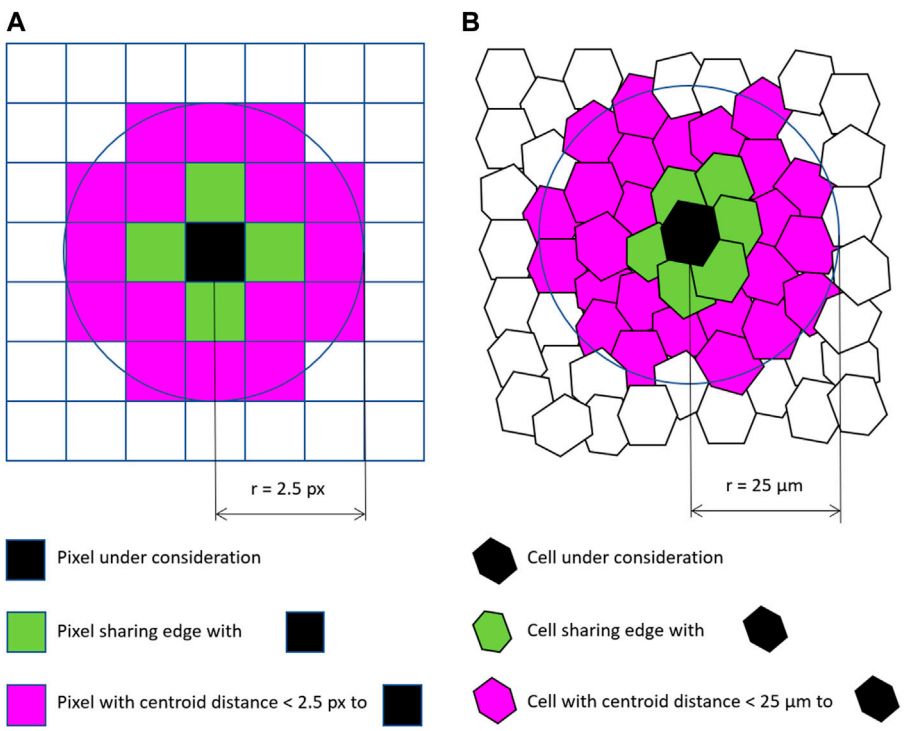


FIGURE 1 | Neighborhood relationships of pixels **(A)** and cells **(B)**: Pixels and cells sharing edges can be considered "touching" neighbors. Cells and pixels with a centroid distance below a given threshold can be considered "proximal" neighbors.

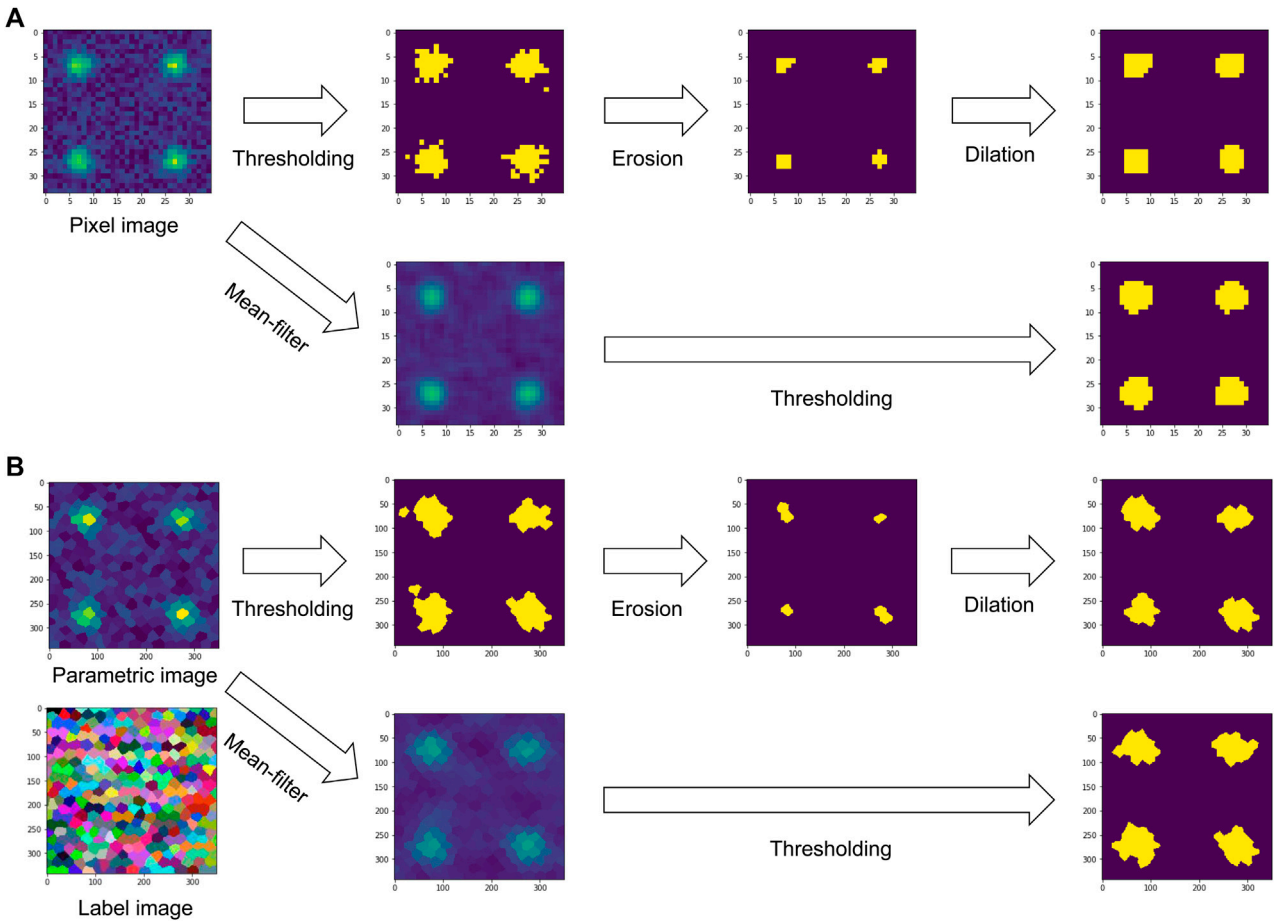


FIGURE 2 | Comparison between filters for binarizing an image on a grid of pixels **(A)** and a grid of cells **(B)**. Both can be used analogously: When applying a mean average filter on a pixel-grid, the intensities of neighboring pixels are averaged. In case of a cell-grid, intensities of cells in a given parametric image are averaged locally to make a new parametric image. The associated label image serves as representation of the region adjacency graph of the cells. Binary image processing filters such as erosion and dilation work analogously, also on the same region adjacency graph expressing neighborhood relationships between cells.

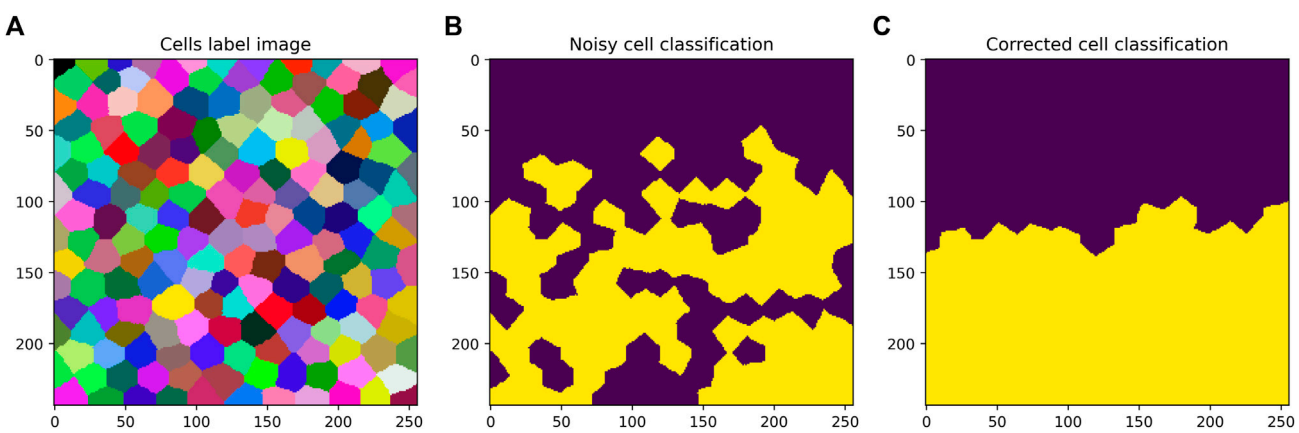


FIGURE 3 | Application of the mode filter to a binary parametric image. Starting from a cell grid shown as label image **(A)** and a corresponding imperfect classification shown as binary parametric image **(B)**, the application of mode filter of proximal cells within a radius of 80 pixels **(C)** can be used to locally correct classification results. Note that the cells are no hexagons. The simulated tissue consists of cells with variable shape, position and size.

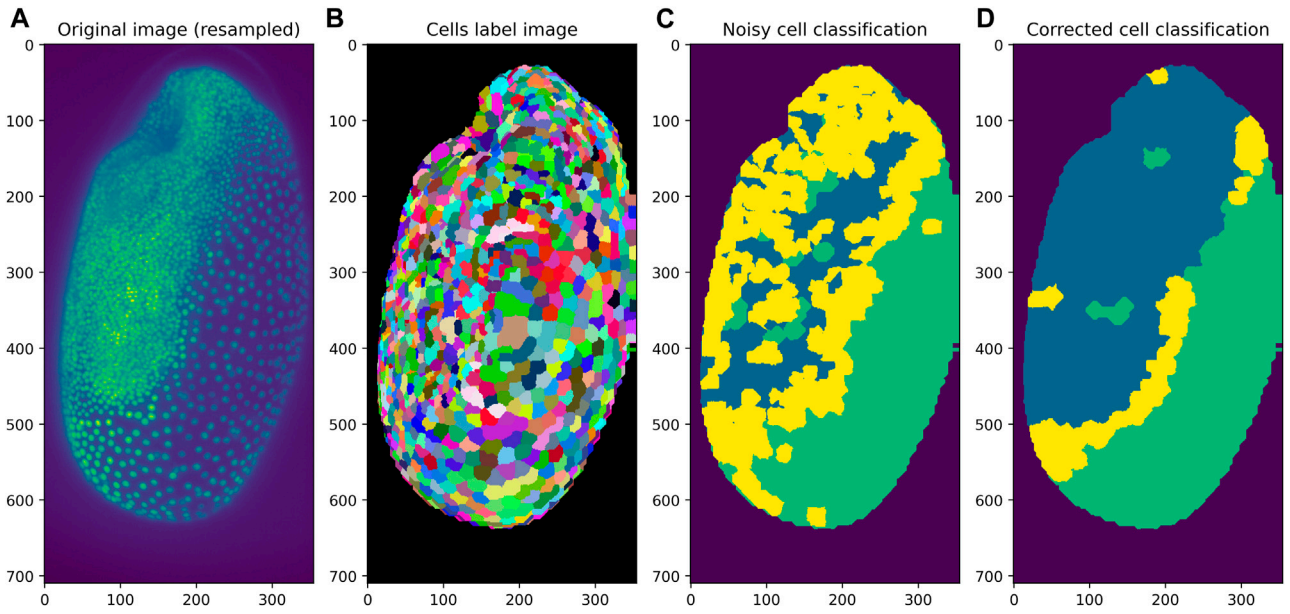


FIGURE 4 | Application of the mode filter to a classification result of cells in a developing *Tribolium castaneum* embryo. Starting from a given 3D image dataset **(A)** and a label image representing the cell segmentation **(B)**, cells can be classified into three classes. If the classification is noisy **(C)**, application of the mode of proximal neighbors filter can be used to determine the most popular class in each cell's neighborhood **(D)**.

2 METHODS

Analogously to image processing filters taking pixel intensities into account, filtering RAGs of cells is feasible. For example, when applying a mean average filter on neighboring cells, the filter could average the intensity, or any other quantitative parameter, of neighboring cells within a certain region and store the resulting value in the corresponding cell of the resulting image. This way you can combine context specific information and any other neighbourhood properties. As for those filters on tissue level the RAG needs to be taken into account, a visual representation of the RAG is required for a user friendly interface. Label images can be used as convenient visual representation of RAGs. Label images are also based on pixel grids and are used to label objects, e.g. with numbers. Thus, in the case of processing images of tissues, a label image could have intensity values numerating the cells. A given pixel intensity 1 means the pixel belongs to cell 1. A label image containing n cells can have pixel intensities ranging from 0 to n, with 0 representing background or empty space between cells. From a label image it is straightforward to retrieve an RAGs following the above specified neighborhood definitions, for example by searching the label image for adjacent pixels with different label values. If there are two neighboring pixels with labels 1 and 2, obviously the cells 1 and 2 are adjacent. In the following, that case will be referred to as "touching neighbors". As mentioned above, touching neighbors are also defined for pixel grids and thus, the analogy between pixel-based image processing filters and cell-based filters is obvious. For determining if cells are within a given circular neighborhood, the centroids of the cells can be used to compute their distances to each other and to derive an RAG based on cell-cell centroid distances below a given threshold. This case can be referred to as "proximal neighbors". Two examples for touching and proximal pixels and cells are given in **Figure 1**. An example workflow applying filters to touching neighbors for the purpose of image segmentation in grids of pixels and grids of cells is shown in **Figure 2**. Furthermore, for the sake of completeness, when studying relationships between segmented objects such as cells, another type of adjacency is commonly used: n-nearest neighbors. Also in this case, centroids of the cells can be used to determine a given number of cells that are adjacent to every cell in a given label image.

The proposed implementation offers so called label neighbor filters for processing labeled objects such as cells. The filters include local minimum, mean, maximum and standard deviation. Those filters work analogously to filters operating on a pixel grid. Furthermore, a filter for calculating the mode of neighbors on a cell grid has been implemented. This filter can be used for post-processing of cell classification results. If a classification algorithm for labeled objects results in a classification that is not perfect and rather noisy, the mode filter can reclassify the cells based on the most popular adjacent cell classification. A synthetical use case for this filter is shown in **Figure 3**.

Parametric images, e.g., showing shape, size and topological parameters of cells in tissues, which can be processed using the proposed filters can be generated using the CLIJ (Haase et al., 2020b) and the MorpholibJ (Legland et al., 2016) libraries in the ImageJ/Fiji ecosystem. Those parametric images are can also be used within the python/napari ecosystem. For generating those parametric images with python, programming experience is necessary, e.g., to make use of the scikit-image (van der Walt et al., 2014) library. Also other quantitative image processing

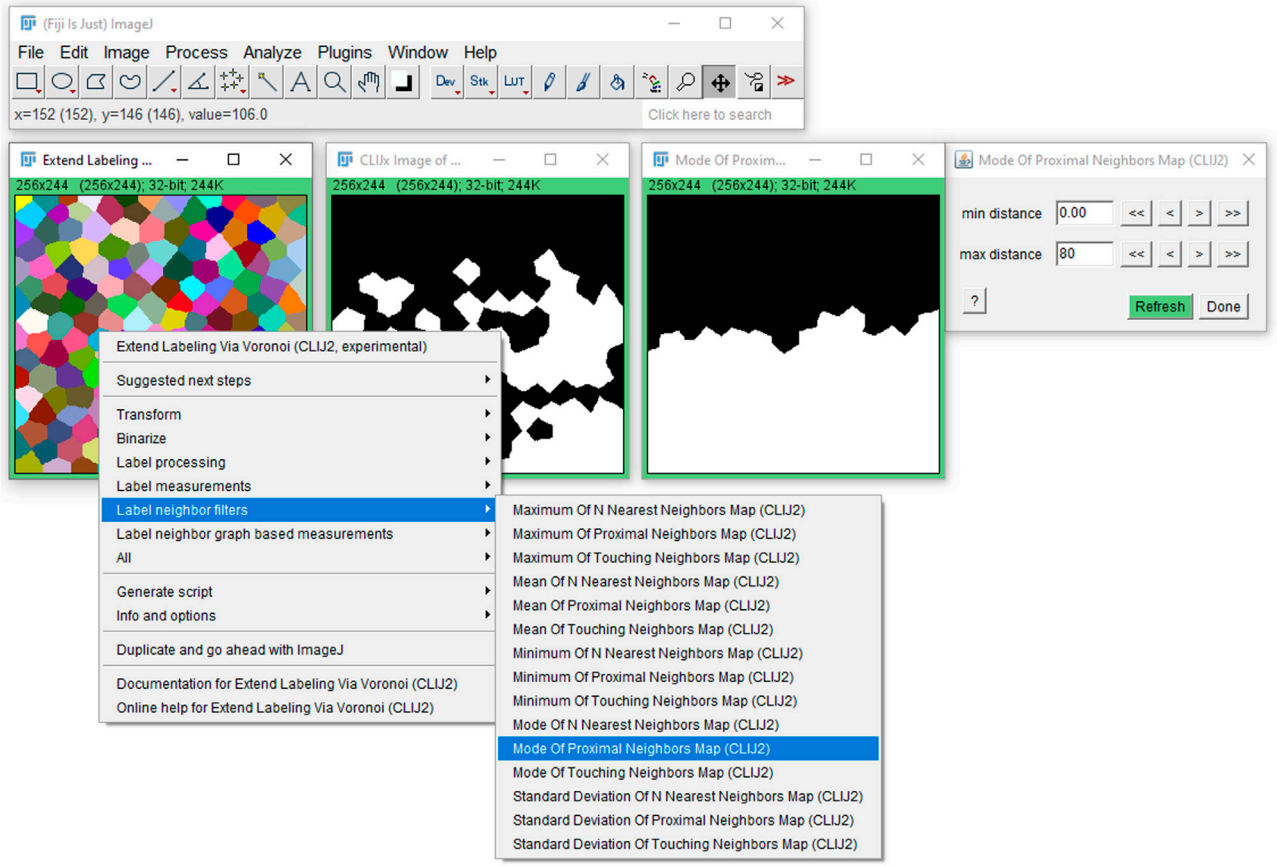


FIGURE 5 | User interface in Fiji: The label neighborhood filters are available in Fiji *via* the CLIJ2-assistant user interface on a right-click. After clicking on the given filter, a new window opens and a corresponding dialog where the user can fine tune parameters such as the maximum distance of cells to be considered in the filter as neighborhood.

libraries can be used. The only constraint is that the parametric images and label images must be available as numpy (Harris et al., 2020) arrays.

To demonstrate the label neighbor filters with a realistic use-case, cell classification in a developing *Tribolium castaneum* embryo serves as example. Starting point is a three dimensional data set of the embryo and a given cell segmentation. A classifier based on average distances to neighboring cells was trained to differentiate embryo, serosa and a border region in between the two where possibly the amnion will form later during development (Benton et al., 2013). As visualized in **Figure 4**, the resulting classification is a bit noisy. After filtering the cells for the mode of proximal classes, the cell classification appears more organized spatially. The border region between embryo and serosa can be identified more clearly, which may be crucial for understanding the differentiation of these tissues biologically.

The proposed user interfaces in ImageJ/Fiji and napari offer a category of "Label neighbor filters" containing the above introduced filters as shown in **Figure 5** and **Figure 6**, respectively. The tools are available to end-users as part of the CLIJ2-assistant (Haase et al., 2020a) in Fiji and the napari-pyclesperanto-assistant (Haase et al., 2021) in napari. In Fiji, the tools can be installed by activating the "clij" and "clij2″ update sites. The current development version of the assistant can be installed by activating the "clijx-assistant" update site. The napari-pyclesperanto-assistant can be installed in napari using napari's the plugin installer. The napari plugin is also under development yet as napari itself is in early developmental stage. Tools that are under development should be used with care and we developers highly value feedback to guide development of our tools in the right direction. Both, the Fiji and the napari plugins, execute the filters based on GPU-accelerated image processing kernels of the CLIJ library. The filters are available in an interactive fashion allowing the users to change parameters, such as the maximum distance of cells to be considered as proximal neighbors, and see the resulting filtered parametric image in real time. Furthermore, in both platforms the user can generate code that represents a workflow as designed on screen and thus, enable reproducibility of the workflow incorporating label neighbor filters potentially together with classical image processing, segmentation, analysis and visualisation steps.

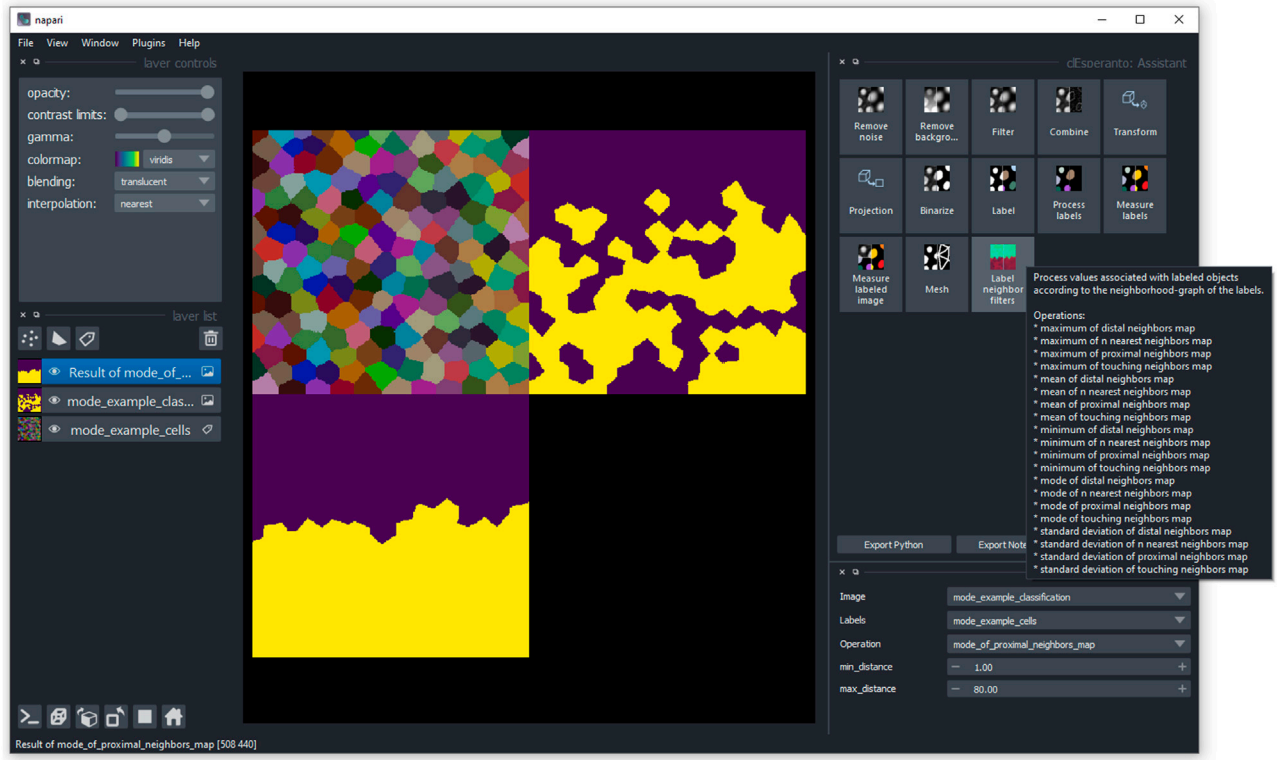


FIGURE 6 | User interface in napari: Users can access the label neighborhood filters *via* the napari-pyclesperanto-assistant user interface. After clicking on a category, the user can select a filter such as mode of proximal neighbors and fine tune parameter such as the maximum distance of cells to be considered for determining the mode.

3 DISCUSSION

The presented label neighbor filters fill a gap in the landscape of available image processing techniques and build a bridge between biologists, bio-image analysts and graph/network scientists. When studying cells and tissues using computational methods, interdisciplinary communication and terminology has to be chosen carefully. Typically, experts studying measurements of cells on tissue level speak about RAGs. Those graphs consist of nodes, which represent cells, and edges which connect nodes and thus, represent the adjacency/neighborhood relationship between cells. Edges could also be seen as abstract representation of tight junctions between two cells. Thus, cell-cell communication, via chemical signals or physical forces, happens along the edge in computational models. While understanding the concepts of graphs, nodes and edges appears crucial from a computer science perspective, a biologist may think more about how cells exchange proteins through their membranes. A physicist may think about forces along cell membranes. Thus, it appears worthwhile to remove any kind of field-specific terminology from a given user interface and present the filters for processing cells and tissues in a generic fashion. As the concepts also work for objects that are no cells, the presented user interfaces use the term "label images" as it is common in the image processing context. Furthermore, while the proposed implementation of the label neighbor filters is limited to basic descriptive statistics operations, the framework is extensible *via* the ImageJ2 plugin mechanism (Rueden et al., 2017) in Fiji. A solution for extending the napari infrastructure is on the way. Contributions to the underlying open source projects are welcome.

When it comes to storing information such as neighborhood relationships, choosing the right file format is crucial. NetworkX supports a JSON-based file format, which is not compatible to any ImageJ/Fiji or napari plugin at the time of writing this. On Fiji side, plugins for managing lineage trees, which are in practice spatio-temporal RAGs, can be managed using plugins such as Trackmate (Tinevez et al., 2017) and stored as XML-based files. Due to the lack of an accepted community standard for cross-platform file formats for storing RAGs, and no available interactive graphical user interfaces for managing those in the biological image processing context, the proposed software uses label images as interchange format for RAGs. Label images can be stored in manifold file formats such as TIF and manipulated using various software packages including plugins for ImageJ, Fiji and napari. The additional processing step to retrieve an RAG from the label image can be neglected in our implementation as processing is executed on graphic processing units (GPUs). This operation can be parallelized and thus, fits well in the scope of GPU-acceleration libraries such as CLIJ.

4 CONCLUSION

In this manuscript I presented filters for local filtering of quantitative parameters of segmented and labeled cells within their local

environment in two and three dimensional imaging data. Such tools appear crucial for visualization, analysis and deeper understanding of quantitative parameters on tissue level. Special emphasis was put on accessibility and user-friendliness to enable a broad audience to explore their image data of cells and tissues.

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 below: https://github.com/haesleinhuepf/label_neighbor_filters.

AUTHOR CONTRIBUTIONS

RH developed the software, made the figures and wrote the manuscript.

FUNDING

I acknowledge support by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy—EXC2068–Cluster of Excellence Physics of Life of TU Dresden.

ACKNOWLEDGMENTS

I would like to thank Daniela Vorkel, MPI-CBG/CSBD Dresden, for imaging the used Tribolium embryo data and Nicola Maghelli and Gene Myers, MPI-CBG/CSBD Dresden, for providing the imaging infrastucture. Furthermore, I thank Christopher Schmied, Human Technopole Milan, and Pradeep Rajasekhar, WEHI Melbourne, for reading the manuscript and providing helpful feedback. Most importantly, I thank Walter Pätzold, HTW Dresden, because he gave an inspring lecture about the topic 15 years ago.

REFERENCES

Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., et al. (2017). Qupath: Open Source Software for Digital Pathology Image Analysis. *Sci. Rep.* 7, 16878. doi:10.1038/s41598-017-17204-5

Benton, M. A., Akam, M., and Pavlopoulos, A. (2013). Cell and Tissue Dynamics during tribolium Embryogenesis Revealed by Versatile Fluorescence Labeling Approaches. *Development* 140, 3210–3220. doi:10.1242/dev.096271

Bhide, S., Gombalova, D., Mönke, G., Stegmaier, J., Zinchenko, V., Kreshuk, A., et al. (2020). Mechanical Competition Alters the Cellular Interpretation of an Endogenous Genetic Programme. *bioRxiv*. 220. e202104107. doi:10.1101/2020.10.15.333963

Haase, R., Jain, A., Rigaud, S., Vorkel, D., Rajasekhar, P., Suckert, T., et al. (2020a). Interactive Design of Gpu-Accelerated Image Data Flow Graphs and Cross-Platform Deployment Using Multi-Lingual Code Generation. *bioRxiv*. doi:10.1101/2020.11.19.386565

Haase, R., Lambert, T., and Kiggins, J. (2021). *Napari-Pyclesperanto-Assistant*, 1. doi:10.5281/zenodo.5501317

Haase, R., Royer, L. A., Steinbach, P., Schmidt, D., Dibrov, A., Schmidt, U., et al. (2020b). Clij: Gpu-Accelerated Image Processing for Everyone. *Nat. Methods* 17, 5–6. doi:10.1038/s41592-019-0650-1

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). "Exploring Network Structure, Dynamics, and Function Using Networkx," in Proceedings of the 7th Python in Science Conference. Editors G. Varoquaux, T. Vaught, and J. Millman (, CA USA), 11–15.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array Programming with NumPy. *Nature* 585, 357–362. doi:10.1038/s41586-020-2649-2

Legland, D., Arganda-Carreras, I., and Andrey, P. (2016). MorphoLibJ: Integrated Library and Plugins for Mathematical Morphology with ImageJ. *Bioinformatics* 32, btw413–3534. doi:10.1093/bioinformatics/btw413

Münster, S., Jain, A., Mietke, A., Pavlopoulos, A., Grill, S. W., and Tomancak, P. (2019). Attachment of the Blastoderm to the Vitelline Envelope Affects Gastrulation of Insects. *Nature* 568, 395–399. doi:10.1038/s41586-019-1044-3

Rauzi, M., Verant, P., Lecuit, T., and Lenne, P.-F. (2008). Nature and Anisotropy of Cortical Forces Orienting drosophila Tissue Morphogenesis. *Nat. Cel Biol.* 10, 1401–1410. doi:10.1038/ncb1798

Ren, X., and Malik, J. (2003). Learning a Classification Model for Segmentation. *Proc. Ninth IEEE Int. Conf. Comp. Vis.* 1, 10–17. doi:10.1109/ICCV.2003.1238308

Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T., et al. (2017). Imagej2: Imagej for the Next Generation of Scientific Image Data. *BMC Bioinformatics* 18, 529. doi:10.1186/s12859-017-1934-z

Schindelin, J., Rueden, C. T., Hiner, M. C., and Eliceiri, K. W. (2015). The Imagej Ecosystem: An Open Platform for Biomedical Image Analysis. *Mol. Reprod. Dev.* 82, 518–529. doi:10.1002/mrd.22489

Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). Nih Image to Imagej: 25 Years of Image Analysis. *Nat. Methods* 9, 671–675. doi:10.1038/nmeth.2089

Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Yamauchi, K., Solak, A. C., et al. (2020). Napari: Available at: https://zenodo.org/record/4048613. accessed 2020-10-31. doi:10.5281/ZENODO.4048613[Dataset]

Stern, C. D. (2004). *Gastrulation: From Cells to Embryo*. Cold Spring Harbor Laboratory Press.

Stoltzfus, C. R., Filipek, J., Gern, B. H., Olin, B. E., Leal, J. M., Wu, Y., et al. (2020). Cytomap: A Spatial Analysis Toolbox Reveals Features of Myeloid Cell Organization in Lymphoid Tissues. *Cel Rep.* 31, 107523. doi:10.1016/j.celrep.2020.107523

Tinevez, J.-Y., Perry, N., Schindelin, J., Hoopes, G. M., Reynolds, G. D., Laplantine, E., et al. (2017). Trackmate: An Open and Extensible Platform for Single-Particle Tracking. *Methods* 115, 80–90. doi:10.1016/j.ymeth.2016.09.016

Toth, T., Balassa, T., Bara, N., Kovacs, F., Kriston, A., Molnar, C., et al. (2018). Environmental Properties of Cells Improve Machine Learning-Based Phenotype Recognition Accuracy. *Sci. Rep.* 8, 10085. doi:10.1038/s41598-018-28482-y

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., et al. (2014). Scikit-Image: Image Processing in Python. *PeerJ* 2, e453. doi:10.7717/peerj.453

Yang, T., Ozcoban, V., Pasam, A., Kocovski, N., Pizzolla, A., Huang, Y.-K., et al. (2020). Spiat: An R Package for the Spatial Image Analysis of Cells in Tissues. *bioRxiv*. doi:10.1101/2020.05.28.122614

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

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 Computer Science

METHODS
published: 09 December 2021
doi: 10.3389/fcomp.2021.777837

SynActJ: Easy-to-Use Automated Analysis of Synaptic Activity

Christopher Schmied[1,2]*, *Tolga Soykan*[1], *Svenja Bolz*[1], *Volker Haucke*[1]* and *Martin Lehmann*[1]*

[1]*Leibniz-Forschungsinstitut für Molekulare Pharmakologie (FMP), Berlin, Germany,* [2]*Fondazione Human Technopole (HT), Milano, Italy*

Neuronal synapses are highly dynamic communication hubs that mediate chemical neurotransmission *via* the exocytic fusion and subsequent endocytic recycling of neurotransmitter-containing synaptic vesicles (SVs). Functional imaging tools allow for the direct visualization of synaptic activity by detecting action potentials, pre- or postsynaptic calcium influx, SV exo- and endocytosis, and glutamate release. Fluorescent organic dyes or synapse-targeted genetic molecular reporters, such as calcium, voltage or neurotransmitter sensors and synapto-pHluorins reveal synaptic activity by undergoing rapid changes in their fluorescence intensity upon neuronal activity on timescales of milliseconds to seconds, which typically are recorded by fast and sensitive widefield live cell microscopy. The analysis of the resulting time-lapse movies in the past has been performed by either manually picking individual structures, custom scripts that have not been made widely available to the scientific community, or advanced software toolboxes that are complicated to use. For the precise, unbiased and reproducible measurement of synaptic activity, it is key that the research community has access to bio-image analysis tools that are easy-to-apply and allow the automated detection of fluorescent intensity changes in active synapses. Here we present SynActJ (Synaptic Activity in ImageJ), an easy-to-use fully open-source workflow that enables automated image and data analysis of synaptic activity. The workflow consists of a Fiji plugin performing the automated image analysis of active synapses in time-lapse movies *via* an interactive seeded watershed segmentation that can be easily adjusted and applied to a dataset in batch mode. The extracted intensity traces of each synaptic bouton are automatically processed, analyzed, and plotted using an R Shiny workflow. We validate the workflow on time-lapse images of stimulated synapses expressing the SV exo-/endocytosis reporter Synaptophysin-pHluorin or a synapse-targeted calcium sensor, Synaptophysin-RGECO. We compare the automatic workflow to manual analysis and compute calcium-influx and SV exo-/endocytosis kinetics and other parameters for synaptic vesicle recycling under different conditions. We predict SynActJ to become an important tool for the analysis of synaptic activity and synapse properties.

Keywords: pHluorin imaging, calcium imaging, synapse, synaptic activity, ImageJ, Fiji, bioimage analysis, computer vision

OPEN ACCESS

Edited by:
Florian Levet,
Institut Interdisciplinaire de Neurosciences (IINS), France

Reviewed by:
Natali Lujan Chanaday Ricagni,
Vanderbilt University, United States
Bertrand Vernay,
Institut de Génétique et de Biologie Moléculaire et Cellulaire (IGBMC), France

***Correspondence:**
Christopher Schmied
christopher.schmied@fht.org
Volker Haucke
Haucke@fmp-berlin.de
Martin Lehmann
MLehmann@fmp-berlin.de

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *15 September 2021*
Accepted: *19 November 2021*
Published: *09 December 2021*

Citation:
Schmied C, Soykan T, Bolz S, Haucke V and Lehmann M (2021) SynActJ: Easy-to-Use Automated Analysis of Synaptic Activity. Front. Comput. Sci. 3:777837. doi: 10.3389/fcomp.2021.777837

INTRODUCTION

Neuronal synapses are important cellular communication hubs that transmit signals between neurons to enable brain function (**Figure 1A**). The transmission of the signal is mediated by chemical neurotransmitters that are synthesized in the presynaptic neuron and stored in the lumen of synaptic vesicles (SVs). When an action potential reaches the synapse, calcium enters the presynaptic terminal *via* voltage-gated calcium channels and triggers the exocytic fusion of release-ready SVs with the plasma membrane (Sudhof, 2013). The released neurotransmitter molecules rapidly diffuse through the synaptic cleft to activate postsynaptic receptors of the receiving neuron. Depending on its type, the neurotransmitter can either help (i.e. excitatory transmitters) or hinder (i.e. inhibitory transmitters) the postsynaptic neuron from firing its own action potential. At the presynaptic terminal, shortly after their fusion with the plasma membrane, SVs are recycled *via* endocytosis and refilled with neurotransmitter to sustain synaptic transmission (Soykan et al., 2016).

To study these highly dynamic processes that occur on timescales of milliseconds to seconds and to monitor the molecular components that orchestrate them, functional imaging tools are crucial. For instance, the exocytosis and endocytosis of SVs can be tracked using synapto-pHluorins, chimeric proteins comprising a pH sensitive variant of GFP genetically fused to the luminal domain of a SV membrane protein. As the pH of a SV under physiological conditions is acidic (~pH 5.5) the fluorescence of pHluorin is quenched at steady-state (Miesenböck et al., 1998). Upon arrival of an action potential (e.g. electrical stimulation of a neuron) SVs release their neurotransmitter content by fusing with the plasma membrane, thereby exposing their lumen to the neutral pH of the extracellular medium (~pH 7.4), upon which pHluorin becomes strongly fluorescent. Subsequent endocytosis of SV membrane proteins from the plasma membrane and rapid acidification of the endocytic vesicle lumen *via* the vesicular ATPase results in re-quenching of the fluorescence signal elicited from pHluorin molecules (**Figures 1B,C**) (Miesenböck et al., 1998; Wienisch and Klingauf, 2006; Kavalali and Jorgensen, 2014). Other assays for synaptic function include molecular sensors that detect the pre- or postsynaptic influx of calcium (**Figures 1D,E**) (Zhao et al., 2011), as well as reporters that monitor changes in membrane voltage or detect the released neurotransmitters (Lin and Schnitzer, 2016). These tools all have in common that they need to detect fast changes in fluorescence signals in small, often diffraction-limited structures to enable the visualization of specific synaptic functions.

The optical detection of synaptic fluorescence is in general performed on live cell or tissue cultures using widefield microscopy to capture as much signal as possible on large, fast and highly sensitive detectors with a diffraction limited resolution of ~200 nm. In order to reduce bleaching and phototoxicity, the light exposure is kept to a minimum. This produces relatively noisy images with abundant out-of-focus signal and many cellular structures that are irrelevant for the downstream analysis (**Figures 1F–H**). The resulting time-lapse movies are thus hard to analyze *via* automated image analysis workflows that only rely on intensity thresholding and detection, often resulting in the analysis being performed manually (Balaji, 2007; Soykan et al., 2017), using custom scripts not easily usable for a wider research community (Chanaday and Kavalali, 2018), or (i.e. for calcium imaging) *via* advanced software toolboxes that require programming expertise (Kaifosh et al., 2014; Pachitariu et al., 2016; Giovannucci et al., 2019; Robbins et al., 2021) (**Supplementary Table S1**). To eliminate the need for tedious, time consuming and biased manual image analysis, the key challenge is to develop robust and easy-to-use software tools that do not require in-depth expertise in programming and image analysis. These tools need to be implemented rigorously, using scientific software engineering to enable easy, transparent and fully reproducible analysis of synaptic imaging data. In recent years the research community has recognized this need and more of such tools have become available for different applications. For example, EZcalcium has been developed for the analysis of calcium imaging data (Cantu et al., 2020). However, due to its implementation in MATLAB, EZcalcium and similar tools require expensive software licenses, thereby limiting their accessibility.

Here we present SynActJ an easy-to-use, entirely free and open source Fiji plugin (Schindelin et al., 2012a) combined with an R shiny app (R-Core-Team, 2020; RStudio-Team, 2020; Chang et al., 2021) that allows the fully automatic detection and analysis of synaptic activity in time-lapse movies. The segmentation parameters can be easily and conveniently adjusted and verified over an entire dataset after which the image analysis can be performed in batch. It is fully reproducible by allowing the saving and loading of the segmentation parameters. SynActJ is implemented as a Fiji plugin in Java and can thus be easily installed *via* a Fiji update site. For the necessary data processing and quality control of the extracted traces a convenient R Shiny app is provided as well as an example script for more detailed analysis and plotting.

AVAILABILITY AND IMPLEMENTATION

The Fiji plugin can be accessed *via* the Cellular Imaging update site: https://sites.imagej.net/Cellular-Imaging/. The code for the Shiny app can be downloaded from github: https://github.com/schmiedc/SynActJ_Shiny. Documentation for using the plugin and the app can be found here: https://schmiedc.github.io/SynActJ/. A test dataset is made available on Zenodo: https://doi.org/10.5281/zenodo.5644945. The code is distributed free and open-source under the MIT license: https://github.com/schmiedc/SynActJ/blob/master/LICENSE. The source code of the Fiji plugin can be downloaded from github: https://github.com/schmiedc/SynActJ. Contact and support: https://forum.image.sc/u/schmiedc/.

RESULTS

Typically, live cell neuronal cultures are imaged using a widefield system with a large field of view. The tissue is illuminated with as

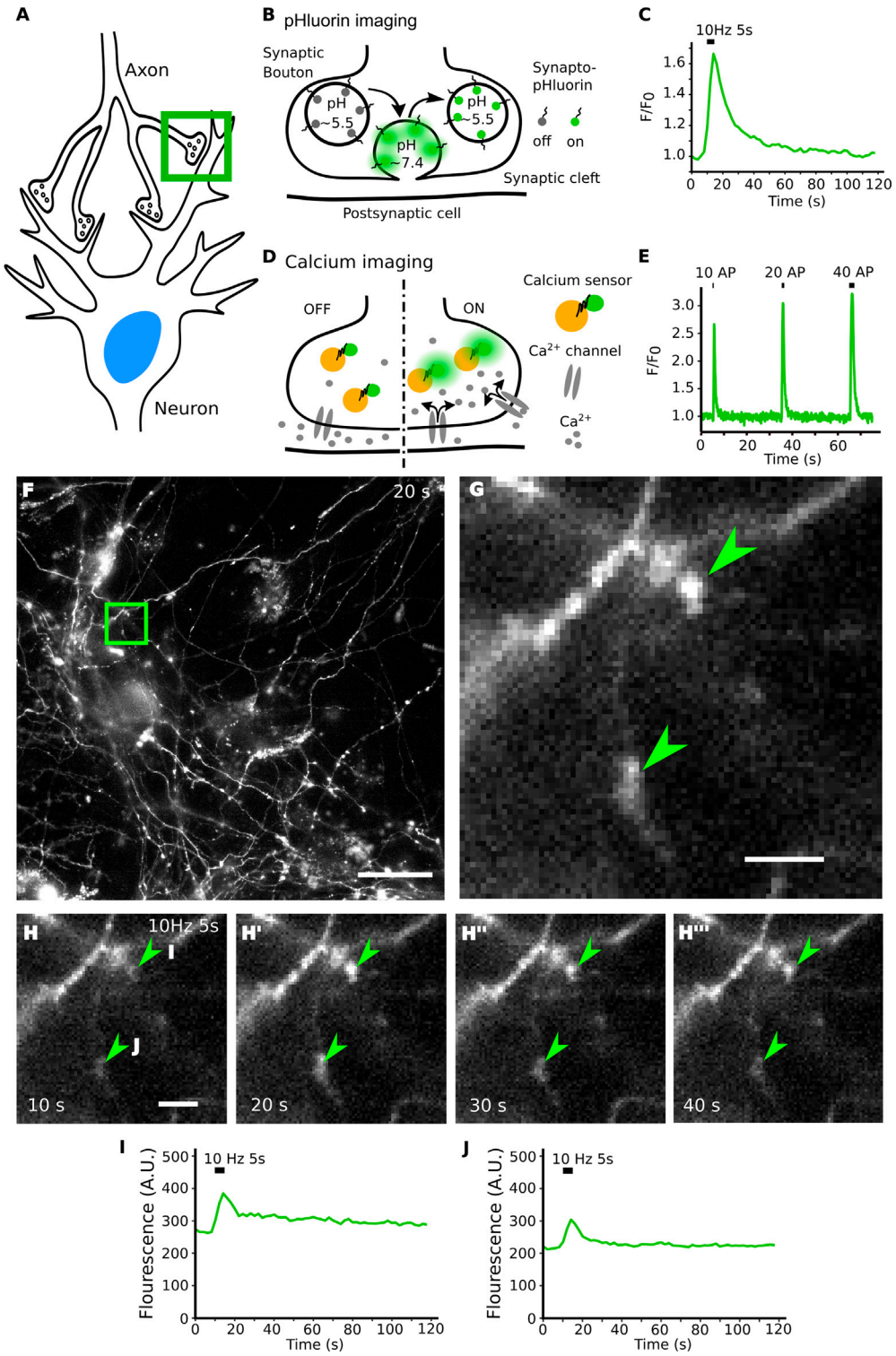


FIGURE 1 | Live imaging of synaptic activity. **(A)** Synaptic signal transmission between neurons depends on the function of synaptic vesicles and on the activity of ion channels. **(B)** Functional imaging by synapto-pHluorin allows studying synaptic vesicle exocytic fusion and subsequent endocytic recycling and acidification using a pH sensitive GFP genetically fused to the luminal domain of a synaptic vesicle membrane protein. The GFP fluorescence is quenched under physiological steady-state conditions in the acidic vesicle lumen. **(B,C)** Upon fusion pHluorin is exposed to the higher extracellular pH, becomes fluorescent and is rapidly quenched when the vesicle is endocytosed, and its lumen returns to its normal acidic pH. **(D,E)** The influx of calcium elicits synaptic vesicle exocytosis and can be detected using calcium sensors. **(F)** A single frame from a full field of view time-lapse movie of a hippocampal neuronal culture transfected with Synaptophysin-pHluorin. [**(G)**, arrowheads]

(*Continued*)

FIGURE 1 | Acquisition with fast and sensitive widefield live imaging using low light exposure leads to images with many irrelevant and out of focus structures around the objects of interest. **(H–H‴)** The increase in fluorescence of Synaptophysin-pHluorin after electrical stimulation is small in size and intensity. **(I,J)** Raw individual traces shown from the top ROI and bottom ROI in panel H. Peak intensity is relatively small over background signal and varies between ROIs. Scale bars: 50 µm (F) and 5 µm **(G,H)**.

little light as possible, to reduce bleaching and phototoxicity. The data sets, while rich in quantitative data, contain out-of-focus signal and many different biological structures that are irrelevant to the analysis (**Figures 1F,G**). For synapto-pHluorin imaging, upon applying field stimulation to the live neuron culture with an electric current, the synaptic boutons respond with a sharp elevated fluorescence signal corresponding to the exocytic fusion and subsequent release of neurotransmitter. Following stimulation, the signal rapidly drops back to baseline on a timescale of seconds as the pHluorin tag gets quenched when SVs are endocytosed and acidified (**Figures 1H–H‴**). These events of briefly elevated signal with relatively small responses over the baseline signal are hard to detect by eye and require that brightness contrast settings are set correctly for perceiving the specific signal responses (**Figures 1I,J**).

SynActJ achieves accurate automatic segmentation of active synaptic boutons (**Figures 2A,B**). It solves the segmentation problem by carving out the portion of the image signal that increases around the time of stimulation (**Figure 2C**). This is achieved by dividing the maximum intensity projection of 10 frames post-stimulation by the maximum intensity projection of the frames before the stimulation. This allows the program to selectively detect responding boutons within the image while removing any non-relevant signal from the remainder of the field of view (**Figures 2E,F**). In order to improve detection of weakly responding synapses an ImageJ implementation of the Laplacian-of-Gaussian (LoG) filter (Sage et al., 2005) is applied to enhance blob-like structures of a defined size range followed by a maximum detection to spot individual events within the diffraction limited resolution of ~200 nm (**Figure 2G**). To create an accurate segmentation of the bouton area, an intensity threshold is applied. The detection and segmentation masks are then used in a marker-controlled watershed using the MorpholibJ Fiji plugin (Legland et al., 2016). This limits the segmentation to objects that have passed the detection threshold and enables the separation of touching detected structures (**Figure 2H**). The segmentation can be further refined by performing a size and object circularity filter. To quantitatively measure the background the maximum intensity projection of the movie is filtered with a large Gaussian blur and an intensity threshold is used to segment the broad signal of the tissue. The resulting mask of the field of view is finally inverted to create a segmentation of the background.

SynActJ provides a graphical user interface which enables the user to interactively adjust the parameters of the workflow *via* a preview (**Figure 2D**). Importantly the user has access to the entire dataset and, thus, is able to quickly verify and adjust the parameters on any of the movies the analysis is applied to. SynActJ ensures full reproducibility of the workflow by allowing the user to conveniently save, load and access all the processing settings. Once suitable segmentation parameters have been defined by the user, the entire image analysis can be executed on the dataset in batch, e.g. different movies of different stimulations or treatments (**Figure 2D**). The segmentation exactly outlines the area of the signal response after stimulation (**Figures 2A,B,H**) and the raw average signal is measured over time in this area. Since the segmentation is outlining the bouton, it is vital that any drift is corrected prior to segmentation using plug-ins such as Correct 3D drift (Parslow et al., 2014).

The resulting measurements are saved in the specified output folder and can then be processed easily using the R Shiny app provided (**Figure 3A**). The app allows rapid processing of the individual traces extracted by the image analysis and provides overview plots for quick and efficient quality control of the data. For each movie the number of ROIs, area of segmented structures and background are plotted (**Figure 3B**). Furthermore, for each movie the raw traces, raw background signals, as well as the average raw traces and background values are plotted and provided as a data table (**Figure 3C**). The raw values can be further filtered and processed in custom data analysis workflows.

The Shiny App also performs basic as well as advanced data processing on each individual movie. The average background value is subtracted from the average signal (**Figure 3D**). To compare the differences in peak responses to the stimulation, the background-corrected signal is surface-normalized by dividing the signal by the average value of the trace in the frames prior to stimulation (**Figure 3E**). To compare how fast the signal returns to baseline a further normalization to the peak of the trace is performed (**Figure 3F**). The result of this data processing is again provided as a data table accessible for further custom statistical analysis and plotting. For instance in order to characterize and quantitatively compare the rate of endocytosis one can fit an exponential decay function to the peak-normalized data and compute the time constant τ (tau: time it took the signal to return to 1/e of the peak) (**Figure 3F**).

To benchmark our workflow we applied SynActJ to an existing dataset that has previously been analyzed manually (Soykan et al., 2017). In this dataset, cultured hippocampal neurons expressing Synaptophysin-pHluorin were treated with DMSO or para-nitro-Blebbistatin (Kepiro et al., 2014), a specific inhibitor for Myosin-II, and stimulated with 50 APs (10 Hz, 5 s) to determine whether Myosin-II activity is involved in SV exo-/endocytosis. In line with the previously proposed role for Myosin-II at synapses (Chandrasekar et al., 2013; Flores et al., 2014; Yue and Xu 2014), para-nitro-Blebbistatin treatment led to reduced Synaptophysin-pHluorin exocytosis after stimulation, as well as delayed endocytosis. Our automated workflow was able to segment more boutons compared to the previously performed manual analysis, which was limited to about 20 ROIs for

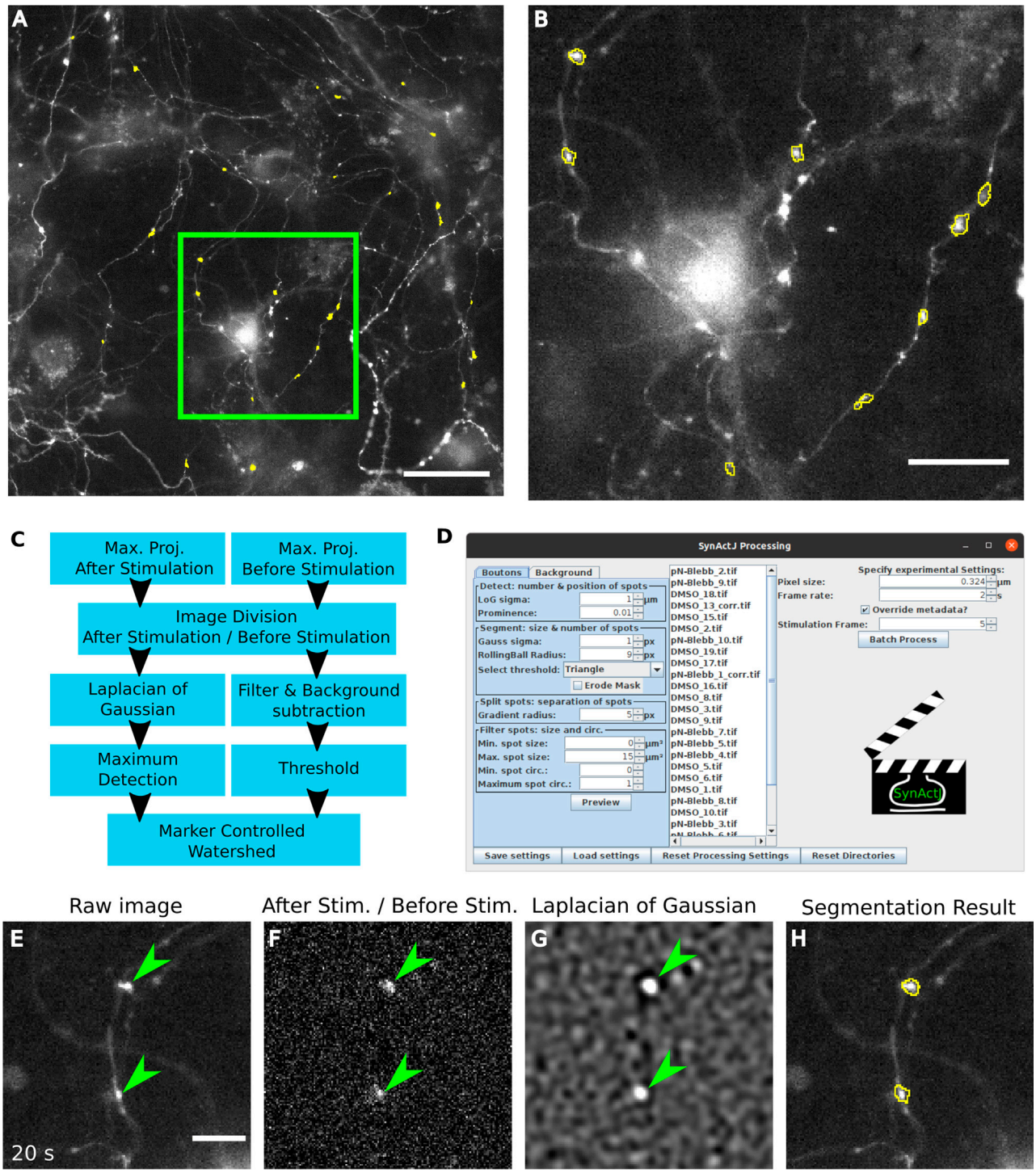


FIGURE 2 | Fast, automatic and easy-to-use image analysis of synaptic activity using SynActJ. **(A)** Hippocampal neuronal cultures transfected with Synaptophysin-pHluorin. [**(A,B)**, yellow outlines] Structures showing a signal response upon stimulation, corresponding to active synaptic boutons are segmented with SynActJ. **(C)** Segmentation is achieved by detecting blob-like structures that display an increase in signal after the stimulation frame and by separating touching structures using a marker-controlled watershed. **(D)** The segmentation parameters can be adjusted and verified on the entire dataset using a GUI that allows testing each segmentation parameter easily, after which the workflow can be executed in batch. [**(E)**, arrowheads] Crop of a raw image showing Synaptophysin-pHluorin with peak fluorescence after electrical stimulation. [**(F)**, arrowheads] The increase in signal around the time point of stimulation is enhanced by dividing the maximum projection of time points after the stimulation by the time points before the stimulation. [**(G)**, arrowheads] Blob-like structures of a specific size are enhanced using a LoG filter]. [**(H)**, yellow outlines] Active boutons are segmented using an intensity threshold and touching objects split with a seeded watershed. Scale bars: 50 µm **(A)**, 20 µm **(B)** and 10 µm **(E–H)**.

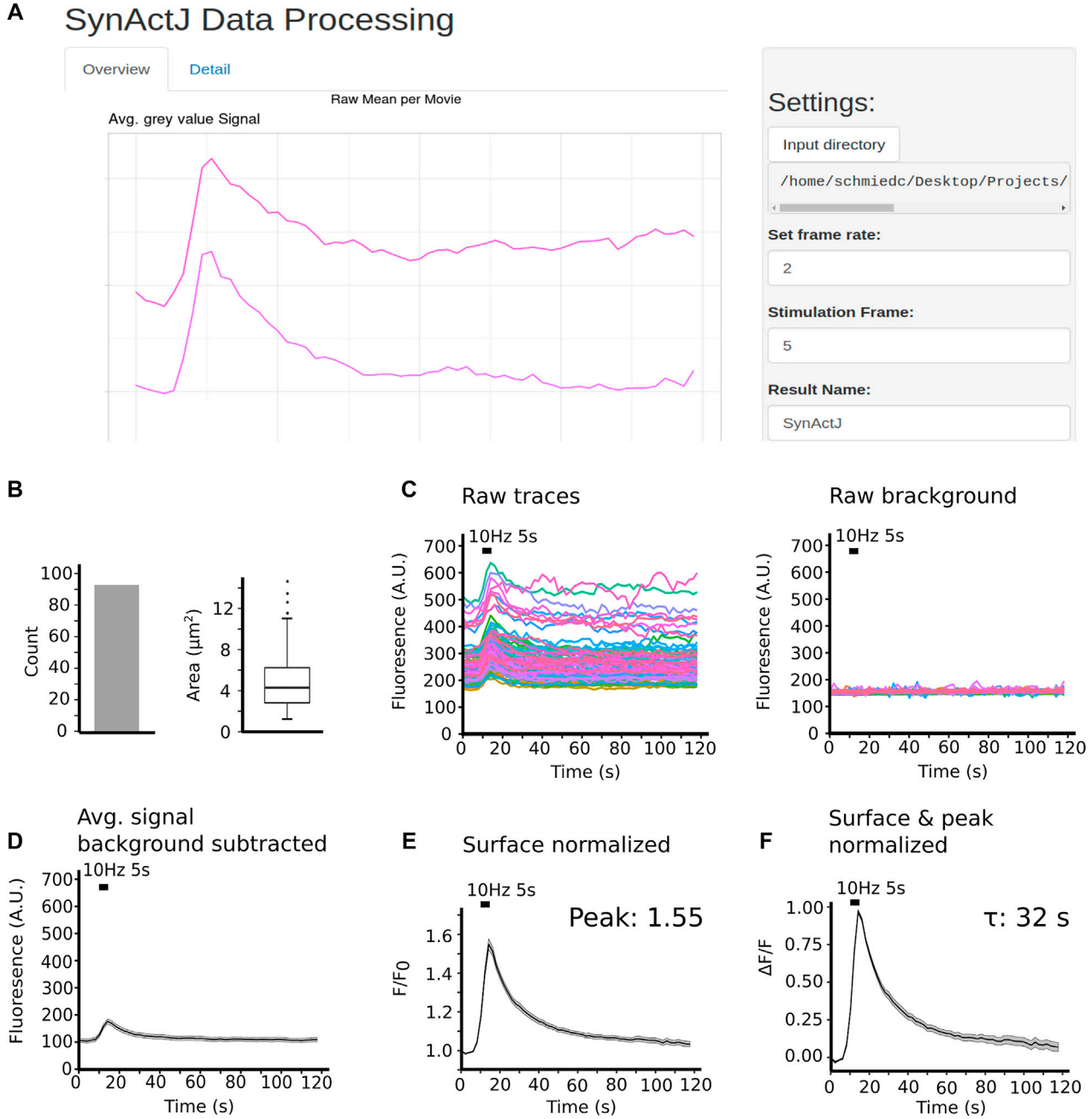


FIGURE 3 | SynActJ enables easy and rapid data processing and analysis *via* an R Shiny app. **(A)** The shiny app allows users to process and plot the result of the image analysis using a GUI. **(B)** Vital parameters for quality control such as number of ROIs and mean area of ROIs per movie are plotted per time-lapse. **(C)** Raw traces of the signal as well as the background are extracted and plotted over time. **(D)** Per movie the average signal is computed, and the background value is subtracted. The average and background subtracted signal traces are surface **(E)** as well as peak normalized **(F)**.

responding boutons and five background ROIs (**Figure 4A**). For further validation we asked how many of the manually selected ROIs also contain an automatically detected ROI. We found that 64% of the manually selected ROIs overlapped with the automatically detected ones (**Figure 4B**). We further determined which objects the automatic analysis missed in the manual dataset and found that many of those structures were large objects such as clusters of boutons or large uniform, potentially out of focus objects (**Figure 4C**). Overall the automatic segmentations of SynActJ seemed to be more restrictive in terms of object size and shape compared to the manual segmentation. To remove false positive automatic detections that by chance showed an increase around the stimulation frame, we removed traces in which the response

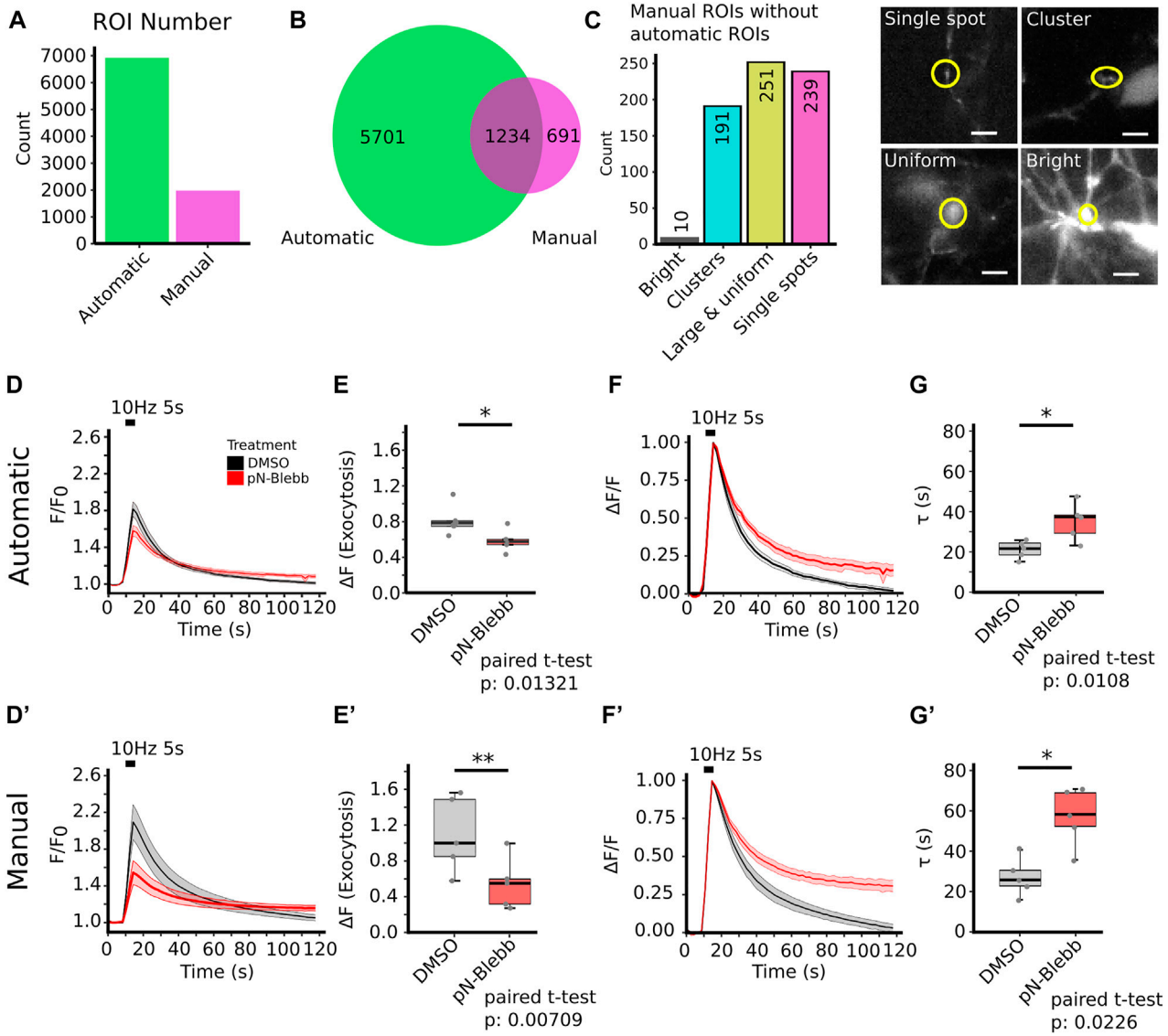


FIGURE 4 | SynActJ allows the rapid analysis of synaptic activity using synapto-pHluorin imaging and reproduces the results of a comparable manual analysis. **(A)** The automatic image segmentation allows extracting many more ROIs from the image data compared to a manual segmentation. **(B)** 64% of the manual ROIs have been detected in the automated analysis. **(C)** Many of the manual ROIs without automatic ROIs are clusters of boutons or relatively large, potential out of focus objects, aside from single boutons (false negative automatic segmentation), example images of these cases are provided. [**(D,E)** compare to **(D′,E′)**] The automatic image analysis is able to reproduce the results from a manually performed image analysis (Soykan et al., 2017) showing significant decrease in peak signal increase after stimulation. **(F,G)** compared to **(F′,G′)** A significant delay of SV endocytosis and reacidification can be detected in the automatic and manual analysis. Data are presented as mean ± standard error of the mean (SEM) in all panels; n. s not significant; $^{*}p < 0.05$; $^{**}p < 0.01$. N = 5 independent experiments; paired *t*-test. Scale bar corresponds to 5 µm **(C)**.

was below two standard deviations of the variation of the background. Further, we removed traces with peaks much later than the stimulation frame. This filter operations removed 312 traces from the original set, leaving 6,623 traces from the automatic segmentation that conform to the necessary response upon stimulation.

The automatic analysis was able to reproduce the results of the manual analysis: Surface-normalized plots revealed a decreased exocytic response to stimulation (**Figures 4D,D′**) while peak-normalized signals displayed a delay in the return to baseline (**Figures 4F,F′**). The automatic analysis also showed a reduced peak signal (**Figures 4E,E′**) and a significant increase in the time constant for returning to baseline intensity upon para-nitro-Blebbistatin application (**Figures 4G,G′**). Interestingly, automatic analysis detected a lower peak signal than analysis by manual assessment. This likely is attributed to the fact our automated analysis workflow is superior with respect to the detection of weakly responding synaptic events, thereby increasing the accuracy of the analysis and reducing bias towards larger and strong responding structures (**Figure 4C**).

TABLE 1 | Signal increase after stimulation.

	Analysis	Mean	Sd	SEM	N
DMSO	automatic	0.82	0.176	0.079	5
pN-Blebb	automatic	0.58	0.128	0.057	5
DMSO	manual	1.10	0.427	0.191	5
pN-Blebb	manual	0.54	0.292	0.131	5

TABLE 2 | Endocytic time constant measurement.

	Analysis	Mean	Sd	SEM	N
DMSO	automatic	29.8	4.44	1.99	5
pN-Blebb	automatic	35.1	9.52	4.26	5
DMSO	manual	27.0	9.36	4.18	5
pN-Blebb	manual	57.5	14.40	6.46	5

This is particularly overt for the DMSO control, for which manual detection overall used much fewer active boutons (**Figures 4E,E′**). Finally, the variability in the extracted parameters is greatly reduced with the standard deviation dropping by a factor of 1.5–2.4 (**Figures 4E,G**; **Tables 1**, **2**), further illustrating the increased precision of the automated approach.

To test if our plugin can be applied to other modes of functional imaging of synaptic activity, we further validated our approach on calcium imaging data. Calcium imaging suffers from similar challenges as those seen for Synapto-pHluorins such as relatively noisy and background-rich live images with fast, yet comparably small fluorescence signal changes over baseline (**Figure 5A–C**). We recorded time-lapse movies of cultured hippocampal neurons expressing the presynaptic calcium sensor Synaptophysin-RGECO and stimulated successively with 10, 20 and 40 APs (all at 40 Hz and 0.25, 0.5 and 1 s, respectively). Similar to the data from Synapto-pHluorin imaging experiments, we were able to rapidly produce accurate segmentations of active boutons in which calcium influx was revealed by RGECO (**Figures 5A,B**). Automated image analysis by SynActJ successfully extracted traces thereby enabling the detection of calcium spikes with overall results near-identical to those produced by manual analysis (**Figures 5D,E**), yet in only a fraction of the time.

DISCUSSION

Biological image data is notoriously difficult to robustly analyze automatically. Not only do the biological and experimental variability hamper computational analysis but also noise and background introduced by hard limitations in the acquisition of biological data. Thus, it is often easier for highly trained scientists to perform these image analysis tasks by hand. For analyzing synaptic activity another key challenge are the brief changes in fluorescent intensity, limited to many small structures throughout a large field of view with heterogeneous signal. This renders manual detection also biased towards strong and large structures that are easier to detect. The difficulty to manually detecting active boutons effectively also limits the number and completeness of detected events that can be analyzed (Soykan et al., 2017). Furthermore, manual selection of responding synaptic terminals using circular ROIs does not allow for precisely outlining the boutons, which is advantageous if there is a slight sample drift or tissue movement. However, this leads to the inclusion of some background area in the final quantification making the measurement less accurate.

The automated analysis of SynActJ efficiently overcomes these challenges. As an easy-to-use workflow it allows the robust and transparent application of well-tested classical image analysis on complex time-lapse movies. It is delivered as an easily installed Fiji plugin for the image analysis as well as a Shiny app that are centered on easy-to-use GUIs. The segmentation is interactive and can be fine-tuned using particle size and shape filters. Basic as well as advanced data analysis is provided *via* an R Shiny app. SynActJ thereby replaces the tedious manual analysis, which may be prone to user-dependent bias, increases as demonstrated the overall accuracy and precision of the analysis, and enables a fully reproducible data analysis by allowing a standardized ROI selection. Together with its compatibility with a wide variety of optical sensors, we believe SynActJ will become an important toolset for the analysis of synaptic activity and function.

METHODS

Preparation of Hippocampal Neuronal Cultures

Hippocampal neuronal cultures were prepared by dissection of hippocampi from postnatal mice at p1-3 and grown in MEM medium (ThermoFisher) supplemented with 5% FCS, 0.5 mM L-glutamine and 2% B-27-supplement. For limiting glial proliferation 2 µM AraC was added at 2 days *in vitro* (DIV). The hippocampal neurons were transfected at DIV 7-9 with 6 µg plasmid DNA carrying synaptophysin fused to pHluorin (gift from L. Lagnado, Cambride, United Kingdom) or SypHy-RGECO (Jackson and Burrone, 2016) (Addgene Plasmid #84078), using a Calcium Phosphate transfection kit (Promega).

Live Imaging

Synapto-pHluorin assay and calcium imaging was performed at DIV13-16 by applying electrical field stimulation to the neuronal culture in a RC-47FSLP stimulation chamber (Warner Instrument) and imaging at 37°C in equilibrated and osmolarity-adjusted basic buffer [170 mM NaCl, 3.5 mM KCl, 0.4 mM KH2PO4, 20 mM N-Tris (hydroxyl-methyl)-methyl-2-aminoethane-sulphonic acid (TES), 5 mM NaHCO3, 5 mM glucose, 1.2 mM Na2SO4, 1.2 mM MgCl2, 1.3 mM CaCl2, 10 mM CNQX and 50 mM AP-5, pH 7.4] using an inverted epifluorescence microscopy (Nikon Eclipse Ti) and 40X/NA 1.3 oil-immersion objective. For detecting synapto-pHluorin signal an eGFP filter set F36-526, and a sCMOS camera (Neo, Andor) were used. Image acquisition was performed at 0.5 frames per second with 100 ms exposure. For calcium imaging, a 580 nm LED (pE4000, CoolLED), backilluminated-sCMOS camera (Prime95B, Photometrics), a Penta DAPI/FITC/Cy3/Cy5/Cy7

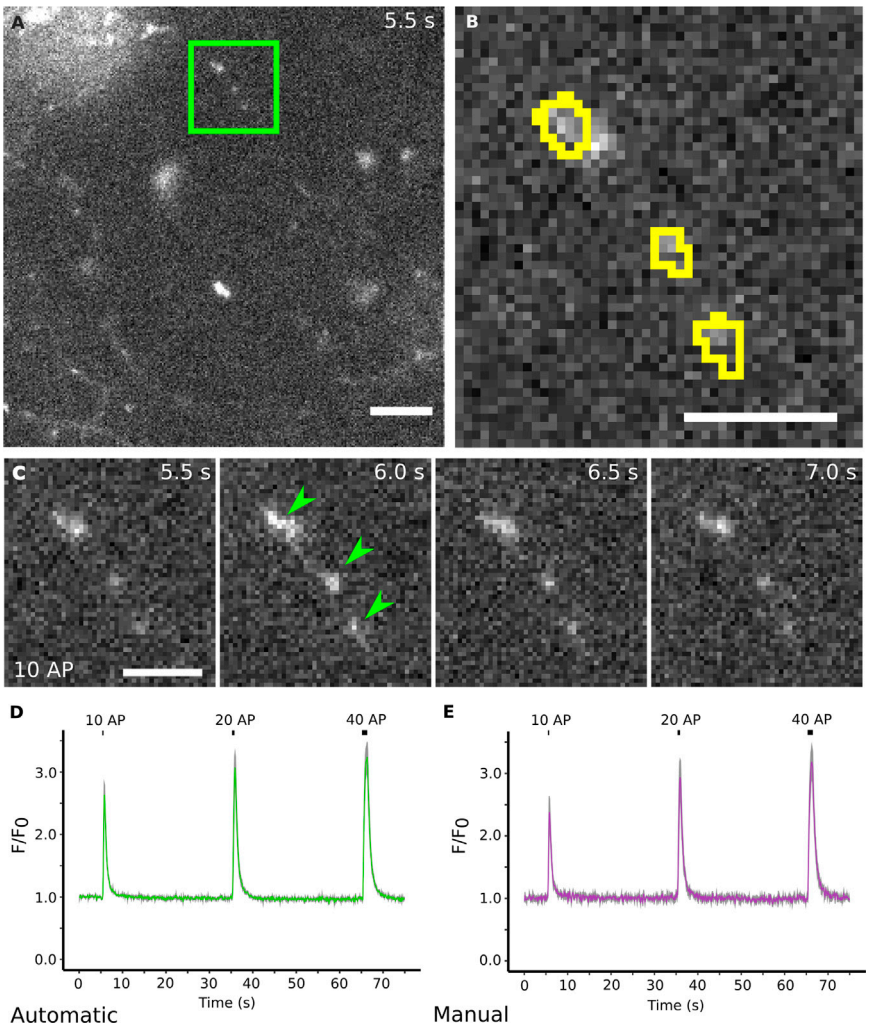


FIGURE 5 | SynActJ can readily be applied to other imaging modalities for synaptic activity such as calcium imaging. **(A)** Cultured mouse neurons transfected with Synaptophysin-RGECO. [**(B)**, yellow outline] Active boutons in calcium imaging data can be successfully segmented]. **(C)** Electric field stimulation of the neurons leads to an increase in fluorescence in synaptic boutons that rapidly drops down to baseline. **(D)** The data analysis allows the visualization of the increase in signal that corresponds to the influx of calcium during/post-stimulation, reproducing the results from manual segmentation **(E)**. Scale bars: 10 µm **(A)** and 5 µm **(B,C)**. Data presented as mean ± SEM. N = 4 images.

Penta LED HC Filter Set (AHF 66-615) and 600/52 emission filter (Semrock) were used. Image acquisition was perfomed at 10 frames per second with 20 ms exposure.

Manual Image Analysis

Manual quantitative analysis of responding boutons was performed in ImageJ (NIH) (Schneider et al., 2012b) by selecting 20 circular ROIs that exhibit remarkable fluorescence increase at the time point of stimulation and five similar-sized circular ROIs selected from the background areas that display uniform fluorescence values throughout the time-lapse image. Examples of these manual regions of interest are provided here: https://doi.org/10.5281/zenodo.5644945. The mean intensity for each ROI is measured using the built-in tools in ImageJ and raw intensity values were calculated by subtracting the average intensity of background ROIs from responding ROIs. Normalized traces were generated for each image by dividing the raw intensity values to the mean intensity before stimulation (for surface normalization) and the maximum intensity (for peak normalization). For the analysis of synapto-pHluorin images, Prism 5 (Graphpad) software were used to determine the post stimulation time constants by fitting the peak normalized values to a mono-exponential decay curve:

$$y_0 + A*e^{-x/t}$$

Constraints: $y_0 = 1$ and zero offset.

Automatic Image Analysis

Movies affected by drift were processed using the Correct 3D drift (Parslow et al., 2014) plugin using multi time scale computation

for detecting slow drifts. Since the tissue presented many edge features, the edge enhancement was turned on. Sub pixel drift correction was turned off to avoid interpolation. Only pixels with gray values larger than 130 were considered. The maximum shift was set to 10 pixel (px) in x and y.

The image analysis plugin was developed as ImageJ1 plugin (Schneider et al., 2012b). To create comparable detections over different image magnification the sigma of LoG is input in micrometre. For the analysis of the synapto-pHluorin time-lapse movies a LoG of 1 µm was used. For the maximum detection a prominence of 0.01 (A.U.) was selected. To segment the area a Gaussian filter with a sigma of 1 px, rolling ball background subtraction of 9 px and an automatic intensity threshold using the Triangle algorithm was used. No erosion was applied to the segmentation masks. For splitting detections that are touching a gradient radius with 5 px radius was applied. The final ROIs were filtered for objects larger than 15 μm^2, no circularity filter was applied. For segmenting the background a Gaussian filter with a sigma of 4 px and an intensity threshold using the MinError algorithm was applied to the maximum projected movie. An example of the raw synapto-pHluorin movies as well as the image analysis parameters and results is provided here: https://doi.org/10.5281/zenodo.5644945.

For the calcium imaging data a LoG sigma of 1 µm and a prominence of 0.001 A.U. For segmenting the area of the boutons a Gaussian filter with sigma of 2 px and a rolling ball background subtraction with radius 50 px was applied. The Triangle algorithm was used for intensity thresholding without applying an erosion to the binary mask. For the seeded watershed a Gradient radius of 10 px was applied. No circularity and size filter was applied. For the background segmentation a Gaussian filter with a sigma of 4 px and the MinError thresholding was applied. The stimulation frame was set to the first stimulation at frame 55.

Automatic Data Analysis

For the synapto-pHluorin datasets the extracted traces per ROI were collected using the SynActJ Shiny app. The detection traces were filtered for false positive detections particularly traces that showed a noisy profile without specific response to the stimulation or traces with linear increase over the movie. These would correspond to false detections in the background that by chance had a sufficiently large intensity difference around the stimulation, bright objects that move slightly in their position over the time-lapse as well as structures in the tissue that increase in intensity without responding to stimulation. For this we computed for each trace an average before the stimulation (6–8 s) as well as at the peak of the intensity just after the stimulation (14–16 s). Traces were removed that had a smaller increase in intensity than 2x the standard deviation of the variation of the background signal. Further, traces were rejected that peaked 26 s after the start of the movie. From the mean traces per movie the mean of the background was subtracted. The background corrected mean over time was then divided by the mean intensity of the frames before stimulation (surface normalization). To determine the exocytic response the peak over the baseline was measured. The surface normalized traces were divided by the peak to perform a peak normalization. Examples for the results of the data processing are provided in the example dataset (https://doi.org/10.5281/zenodo.5644945). For determining the stimulation time constant the same method as the manual image analysis was performed.

For the calcium imaging dataset the extracted traces were collected with the SynActJ Shiny app, no filtering for false positive detections was applied for the further analysis. The mean of the detection traces were subtracted by the mean of the background traces per movie. The corrected mean traces per movie were surface normalized and the means over the different movies plotted as line plot with mean ± SEM.

Comparison of Manual and Automatic ROIs

For testing how many manual ROIs also contain an automatic ROI, a binary mask image was first produced for the automatic segmentations. The manual ROIs were then overlaid onto this binary mask image. Since the manual ROIs do not outline the objects of interest precisely nor they were generated on drift corrected movies, any overlap was counted as manual ROI with an automatic ROI. Manual ROIs with overlap to multiple automatic ROIs was counted only once. Manual ROIs without overlap with an automatic ROI were then further visually analyzed to estimate false negatives. The objects outlined by the manual ROIs were grouped into the following classes: 1. Single bouton. 2. Large structure of cluster of boutons. 3. Very large and uniform, potentially out of focus object. 4. Large and bright structures.

Statistics

All data are presented as mean ± SEM and were obtained from multiple independent experiments (e.g. independent mouse cultures), with total sample numbers provided in the figure legends as N independent experiments. No statistical method was used to pre-determine sample size as sample sizes were not chosen based on pre-specified effect size. Instead, multiple independent experiments were carried out using several sample replicates as detailed in the figure legends. The statistical significance between two groups for all normally distributed data was evaluated with a two-tailed paired Student's t-test.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The animal study was reviewed and approved by the Tierversuchskommission Landesamt für Gesundheit und Soziales (LAGeSo) Berlin.

AUTHOR CONTRIBUTIONS

TS, CS, SB, VH, and ML contributed to conception and design of the project. CS designed and implemented the software. TS and SB performed experiments. SB and TS tested the software. CS, TS, and SB wrote sections of the manuscript. CS and TS performed the data and statistical analysis. All authors contributed to manuscript revision, read, and approved the submitted version.

FUNDING

Supported by grants from the Deutsche Forschungsgemeinschaft (SFB958/A01 to V.H. and HA2686/20-1 to V.H.).

ACKNOWLEDGMENTS

We thank Delia Löwe (FMP-Berlin) for expert technical assistance. Furthermore, we thank Robert Haase (DFG Cluster of Excellence "Physics of Life"/CSBD Dresden) for feedback to the manuscript and valuable material for writing ImageJ plugins.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2021.777837/full#supplementary-material

REFERENCES

Balaji, J. (2007). *Time Series Analyzer*. https://imagej.nih.gov/ij/plugins/time-series.html.

Cantu, D. A., Wang, B., Gongwer, M. W., He, C. X., Goel, A., Suresh, A., et al. (2020). EZcalcium: Open-Source Toolbox for Analysis of Calcium Imaging Data. *Front. Neural Circuits* 14, 25–29. doi:10.3389/fncir.2020.00025

Chanaday, N. L., and Kavalali, E. T. (2018). Optical Detection of Three Modes of Endocytosis at Hippocampal Synapses. *eLife* 7, 1–24. doi:10.7554/eLife.36097

Chandrasekar, I., Huettner, J. E., Turney, S. G., and Bridgman, P. C. (2013). Myosin II Regulates Activity Dependent Compensatory Endocytosis at Central Synapses. *J. Neurosci.* 33 (41), 16131–16145. doi:10.1523/jneurosci.2229-13.2013

Chang, W., Cheng, J., Allaire, J. J., Sievert, C., Schloerke, B., Xie, Y., et al. (2021). *Shiny: Web Application Framework for R*.

Flores, J. A., Balseiro-Gomez, S., Cabeza, J. M., Acosta, J., Ramirez-Ponce, P., and Ales, E. (2014). A New Role for Myosin II in Vesicle Fission. *PLoS One* 9 (6), e100757. doi:10.1371/journal.pone.0100757

Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., et al. (2019). CaImAn an Open Source Tool for Scalable Calcium Imaging Data Analysis. *eLife* 8, 1–45. doi:10.7554/eLife.38173

Jackson, R. E., and Burrone, J. (2016). Visualizing Presynaptic Calcium Dynamics and Vesicle Fusion with a Single Genetically Encoded Reporter at Individual Synapses. *Front. Synaptic Neurosci.* 8, 21. doi:10.3389/fnsyn.2016.00021

Kaifosh, P., Zaremba, J. D., Danielson, N. B., and Losonczy, A. (2014). SIMA: Python Software for Analysis of Dynamic Fluorescence Imaging Data. *Front. Neuroinform* 8, 80. doi:10.3389/fninf.2014.00080

Kavalali, E. T., and Jorgensen, E. M. (2014). Visualizing Presynaptic Function. *Nat. Neurosci.* 17, 10–16. doi:10.1038/nn.3578

Képiró, M., Várkuti, B. H., Végner, L., Vörös, G., Hegyi, G., Varga, M., et al. (2014). para-Nitroblebbistatin, the Non-Cytotoxic and Photostable Myosin II Inhibitor. *Angew. Chem. Int. Ed.* 53 (31), 8211–8215. doi:10.1002/anie.201403540

Legland, D., Arganda-Carreras, I., and Andrey, P. (2016). MorphoLibJ: Integrated Library and Plugins for Mathematical Morphology with ImageJ. *Bioinformatics* 32, 3532–3534. doi:10.1093/bioinformatics/btw413

Lin, M. Z., and Schnitzer, M. J. (2016). Genetically Encoded Indicators of Neuronal Activity. *Nat. Neurosci.* 19, 1142–1153. doi:10.1038/nn.4359

Miesenböck, G., De Angelis, D. A., and Rothman, J. E. (1998). Visualizing Secretion and Synaptic Transmission with pH-Sensitive green Fluorescent Proteins. *Nature* 394, 192–195. doi:10.1038/28190

Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalgleish, H., et al. (2016). Suite2p: Beyond 10,000 Neurons with Standard Two-Photon Microscopy. *bioRxiv*, 061507. doi:10.1101/061507

Parslow, A., Cardona, A., and Bryson-Richardson, R. J. (2014). Sample Drift Correction Following 4D Confocal Time-Lapse Imaging. *J. Vis. Exp.* 12 (86), 51086. doi:10.3791/51086

R-Core-Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robbins, M., Christensen, C. N., Kaminski, C. F., and Zlatic, M. (2021). Calcium Imaging Analysis - How Far Have We Come? *F1000Res* 10, 258. doi:10.12688/f1000research.51755.2

RStudio-Team (2020). *RStudio: Integrated Development Environment for R*. Boston, MA: PBC. http://www.rstudio.com/.

Sage, D., Neumann, F. R., Hediger, F., Gasser, S. M., and Unser, M. (2005). Automatic Tracking of Individual Fluorescence Particles: Application to the Study of Chromosome Dynamics. *IEEE Trans. Image Process.* 14, 1372–1383. doi:10.1109/tip.2005.852787

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012a). Fiji: An Open-Source Platform for Biological-Image Analysis. *Nat. Methods* 9, 676–682. doi:10.1038/nmeth.2019

Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012b). NIH Image to ImageJ: 25 Years of Image Analysis. *Nat. Methods* 9, 671–675. doi:10.1038/nmeth.2089

Soykan, T., Maritzen, T., and Haucke, V. (2016). Modes and Mechanisms of Synaptic Vesicle Recycling. *Curr. Opin. Neurobiol.* 39, 17–23. Elsevier Ltd. doi:10.1016/j.conb.2016.03.005

Soykan, T., Kaempf, N., Sakaba, T., Vollweiter, D., Goerdeler, F., Puchkov, D., et al. (2017). Synaptic Vesicle Endocytosis Occurs on Multiple Timescales and Is Mediated by Formin-Dependent Actin Assembly. *Neuron* 93, 854–866. Elsevier Inc.e854. doi:10.1016/j.neuron.2017.02.011

Südhof, T. C. (2013). Neurotransmitter Release: The Last Millisecond in the Life of a Synaptic Vesicle. *Neuron* 80 (3), 675–690. doi:10.1016/j.neuron.2013.10.022

Wienisch, M., and Klingauf, J. (2006). Vesicular Proteins Exocytosed and Subsequently Retrieved by Compensatory Endocytosis Are Nonidentical. *Nat. Neurosci.* 9 (8), 1019–1027. doi:10.1038/nn1739

Yue, H.-Y., and Xu, J. (2014). Myosin Light Chain Kinase Accelerates Vesicle Endocytosis at the Calyx of Held Synapse. *J. Neurosci.* 34 (1), 295–304. doi:10.1523/jneurosci.3744-13.2014

Zhao, Y., Araki, S., Wu, J., Teramoto, T., Chang, Y.-F., Nakano, M., et al. (2011). An Expanded Palette of Genetically Encoded Ca 2+ Indicators. *Science* 333, 1888–1891. doi:10.1126/science.1208592

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 Computer Science

BRIEF RESEARCH REPORT
published: 04 January 2022
doi: 10.3389/fcomp.2021.796117

ZELDA: A 3D Image Segmentation and Parent-Child Relation Plugin for Microscopy Image Analysis in *napari*

Rocco D'Antuono[1]* *and Giuseppina Pisignano*[2]

[1]Crick Advanced Light Microscopy STP, The Francis Crick Institute, London, United Kingdom, [2]Department of Biology and Biochemistry, University of Bath, Bath, United Kingdom

Bioimage analysis workflows allow the measurement of sample properties such as fluorescence intensity and polarization, cell number, and vesicles distribution, but often require the integration of multiple software tools. Furthermore, it is increasingly appreciated that to overcome the limitations of the 2D-view-based image analysis approaches and to correctly understand and interpret biological processes, a 3D segmentation of microscopy data sets becomes imperative. Despite the availability of numerous algorithms for the 2D and 3D segmentation, the latter still offers some challenges for the end-users, who often do not have either an extensive knowledge of the existing software or coding skills to link the output of multiple tools. While several commercial packages are available on the market, fewer are the open-source solutions able to execute a complete 3D analysis workflow. Here we present ZELDA, a new *napari* plugin that easily integrates the cutting-edge solutions offered by python ecosystem, such as *scikit-image* for image segmentation, *matplotlib* for data visualization, and *napari* multi-dimensional image viewer for 3D rendering. This plugin aims to provide interactive and zero-scripting customizable workflows for cell segmentation, vesicles counting, parent-child relation between objects, signal quantification, and results presentation; all included in the same open-source *napari* viewer, and "few clicks away".

Keywords: image analysis, 3D, segmentation, parent-child, napari, plugin, microscopy, measurement

OPEN ACCESS

Edited by:
Florian Levet,
UMR5297 Institut Interdisciplinaire de Neurosciences (IINS), France

Reviewed by:
Stephane Rigaud,
Institut Pasteur, France
Sebastian Gonzalez-Tirado,
European Molecular Biology Laboratory Heidelberg, Germany

***Correspondence:**
Rocco D'Antuono
rocco.dantuono@crick.ac.uk

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *15 October 2021*
Accepted: *11 November 2021*
Published: *04 January 2022*

Citation:
D'Antuono R and Pisignano G (2022) ZELDA: A 3D Image Segmentation and Parent-Child Relation Plugin for Microscopy Image Analysis in napari. *Front. Comput. Sci. 3:796117. doi: 10.3389/fcomp.2021.796117*

INTRODUCTION

Microscopy and image analysis significantly contribute to the advancement of research in life sciences. However, researchers operating microscopes have to deal with a number of experimental challenges often requiring different types of image analysis procedures. For instance, the counting of protein structures, such as the ProMyelocytic Leukemia Nuclear Bodies (PML NB) found involved in chromatin remodeling, telomere biology, senescence or viral infections (Lallemand-Breitenbach and de The, 2018), is achievable by applying a "2D counting" image analysis tool to first identify cells and then determine the number of contained PML NB (**Supplementary Figure S1A**). Similarly, in experiments where the measurement of transient concentration of Ca^{2+} or metabolites is assessed, a stable staining and reliable segmentation of individual cytoplasmic organelles might be required to then apply a "2D measurement" of fluorescence intensity and organelle shape (**Supplementary Figure S1B**). This can be fundamental in studies of mitochondrial metabolism where a complex correlation between ER-mitochondria Ca^{2+} fluxes and autophagy have been highlighted (Missiroli et al., 2020). Furthermore, some kidney pathological conditions, such as the glomerulocystic disease,

could originate from topological defects acquired during development (Fiorentino et al., 2020). Such conditions can be studied using a staining to identify single cells, glomeruli, and the renal tubular system (**Supplementary Figure S1C**). The conformational study of a glomerulus, with the assessment of the number of cells, is referred to as "3D cell counting" or "3D object segmentation". In influenza infection, instead, the released viral genome can be involved in mechanisms such as replication or viral protein transcription and identified by the presence of a negative-sense RNA (Long et al., 2019). The dynamics of the viral infection can therefore be monitored by localizing the RNA molecules within the cell nuclei (**Supplementary Figure S1D**) in a task definable as "3D object segmentation" and "parent-child relation".

The ability to extrapolate valuable results from microscopy experiments as those just mentioned, mainly relies on the image analysis knowledge, and availability of the right software tools for the specific purpose. The bioimage analysis is a combination of multiple informatics tools (referred to as "components") organized into "workflows" with different levels of complexity (Miura et al., 2020). Such components are often available only by scripting and researchers may struggle to find an effective way of combining them together in a complete workflow. To date, there have been great initiatives to both promote the bioimage analysis (NEUBIAS Training Schools (Martins et al., 2021)) and raise awareness about informatics tools (BioImage Informatics Index, http://biii.eu/), while a growing number of excellent open-source software became available (Schindelin et al., 2012) (McQuin et al., 2018). However, the end-user has still to acquire a minimum level of bioinformatic knowledge in order to analyze image data.

A recent survey proposed by the COBA[1] to the bioimage analysis community has suggested that the most used bioimage analysis tools belong to the category of the "open-source point and click software" and there is a high demand for better software for "3D/Volume" and "Tissue/Histology" analysis (Jamali et al., 2021), underlining the urgency of more and more new, easy and customizable tools for multi-dimensional image segmentation.

Furthermore, to guarantee the experimental reproducibility, minimize the mistakes, and preserve scientific integrity, any new analysis software should include accurate logging of the used parameters at each step of the workflow[2].

To facilitate life science researchers during the application of image analysis to biological experiments, we developed ZELDA: a *napari* plugin for the analysis of 3D data sets with multiple object populations. ZELDA has the advantage of being equipped with ready-to-use protocols for 3D segmentation, measurement, and "parent-child" relation between object classes. It then allows the rapid cell counting, quantification of vesicle distribution, and the fluorescence measurement of subcellular compartments for most biological applications. Since each image analysis workflow is designed as a simple protocol with numbered steps, it requires no knowledge of image analysis and it's sufficient to follow the step-by-step instructions to perform a complete analysis. Furthermore, while the integration in *napari* allows to easily view each step of the image processing as 2D slice or 3D rendering, the visibility, opacity and blending modulation facilitates the tuning of the used parameters (for example threshold value or gaussian filter size) by visualizing multiple layers at the same time.

Altogether ZELDA plugin is a new easy to use open-source software designed to assist researchers in the most common bioimage analysis applications without requiring any scripting knowledge.

MATERIALS AND METHODS

Image Acquisition

The data sets of influenza infected human eHAP cells, BPAE cells (Invitrogen FluoCells Slide #1) and mouse kidney tissue (Invitrogen FluoCells Slide #3), shown and analyzed in (**Figures 2**, **4**, **5**, **Supplementary Figure S1**; **Supplementary Figure S3**) have been acquired with a Zeiss LSM880 confocal microscope, using a Plan-Apochromat 20X/0.8 NA objective. A sequential acquisition for DAPI (excitation 405 nm, detection in the range 420–462 nm), AlexaFluor 488 and AlexaFluor 568 (excitation 561 nm, detection in the range 570–615 nm) was used to acquire z-stacks with the total size up to 13 um, every 0.5 um. Pixel size was 0.20 um.

The beads used to show the segmentation workflow (**Figure 1**) were TetraSpeck™ Microspheres, 0.1 μm; images were acquired on a Zeiss Observer.Z1 using Micro-Manager (https://micro-manager.org/) software with a Hamamatsu ORCA-spark Digital CMOS camera, using a 63X/1.4 NA objective. Pixel size is 0.08 um.

Object Segmentation, Measurements, and Results Export

The segmentation obtained by running the ZELDA protocols is achieved using *scikit-image* (van der Walt et al., 2014) (version 0.18.1) and *SciPy* (Virtanen et al., 2020) (version 1.6.3) modules for image processing in python.

The resulting measurements are handled as *Pandas* data frames (McKinney, 2011) (version 1.2.4) and plotted with *Matplotlib* (Hunter, 2007) (version 3.4.2). JupyterLab Version 3.0.14 was used to handle the result tables (*pandas*), calculate the jaccard index (*scikit-learn* (Pedregosa et al., 2011)), and plot the data (*matplotlib*). Additionally, the latest version of *napari-zelda* uses *datatable* package to handle results (https://github.com/h2oai/datatable).

Graphical User Interface Design, Plugin Development, Installation, and Execution

ZELDA plugin for *napari* ("*napari-zelda*") can be installed through the "Install/Uninstall Package(s)" menu in *napari* (napari contributors, 2019), and its interface can be added with "Plugins/Add dock widget".

[1]Center for Open Bioimage Analysis: https://openbioimageanalysis.org/.

[2]Kota Miura 2020, "In Defense of Image Data & Analysis Integrity" - [NEUBIASAcademy@Home] Webinar: https://www.youtube.com/watch?v=c_Oi2HKom_Y.

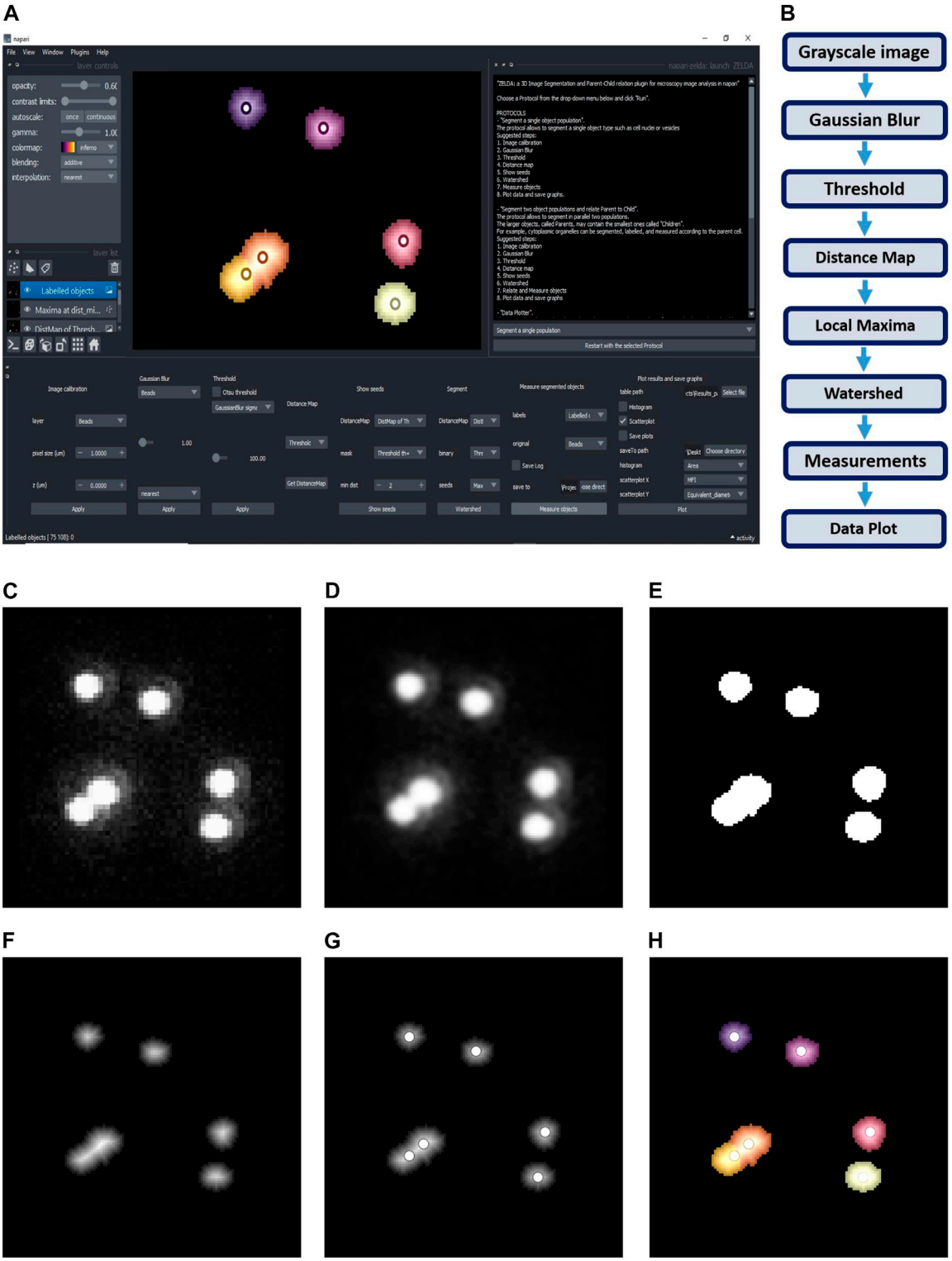


FIGURE 1 | ZELDA plugin for *napari*. **(A)** GUI of ZELDA with the description of the ready-to-use protocols sufficient to run a complete image analysis workflow (white text box on the top right). Each protocol is divided into numbered steps corresponding to the software commands in the *napari* dock widget (bottom of the software interface). **(B)** "Segment a single population" protocol including a minimum number of processing operations. **(C)** Original image. **(D)** Gaussian Blur of the original image. **(E)** Binary image obtained applying a Threshold to the gaussian blur. **(F)** Distance Map applied to the binary image. **(G)** Seeds (Local Maxima) used to run the Watershed. **(H)** Objects labelled by Watershed. The labelled objects can then be measured, and the results exported.

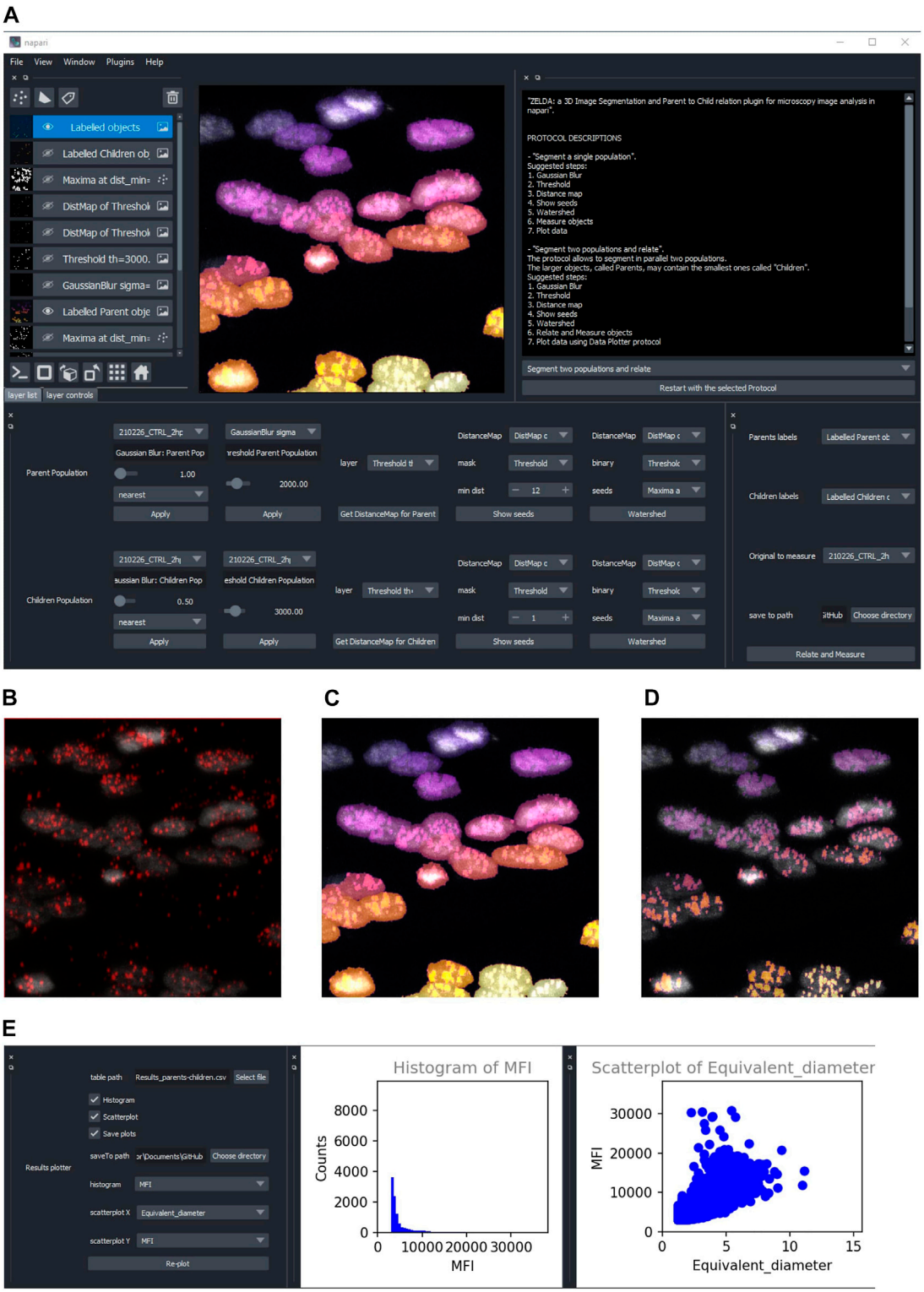


FIGURE 2 | ZELDA application for the 3D segmentation of two object populations and "Parent-child" relation. **(A)** ZELDA protocol "Segment two populations and relate" used to analyze the distribution of viral RNA in infected human cell nuclei. **(B)** Original 3D data set showing a nuclear staining with DAPI (gray) and an RNA staining with AlexaFluor 568 (red). **(C)** The nuclei and the RNA aggregates, individually segmented and **(D)** the RNA aggregates (children population) labelled according to the containing nuclei (parent population). **(E)** Resulting measurements reimported and plotted with the "Data Plotter" protocol.

FIGURE 3 | Design of a custom image analysis workflow with ZELDA without requiring any scripting knowledge. **(A)** Choice of the number of steps for the new protocol to implement a custom image analysis workflow. **(B)** Drop-down menu showing all the modules implemented in ZELDA. **(C)** Assignment of an operation to a specific step of the new protocol. **(D)** Example of updated JSON database that controls the software layout, once a new protocol is saved. **(E)** The newly created protocol GUI available after having restarted ZELDA. **(F)** "Import and Export Protocols" allows the user to import and export the content of the ZELDA .json database. Either a new file is created or protocols are appended to the destination database to easily share it with the community.

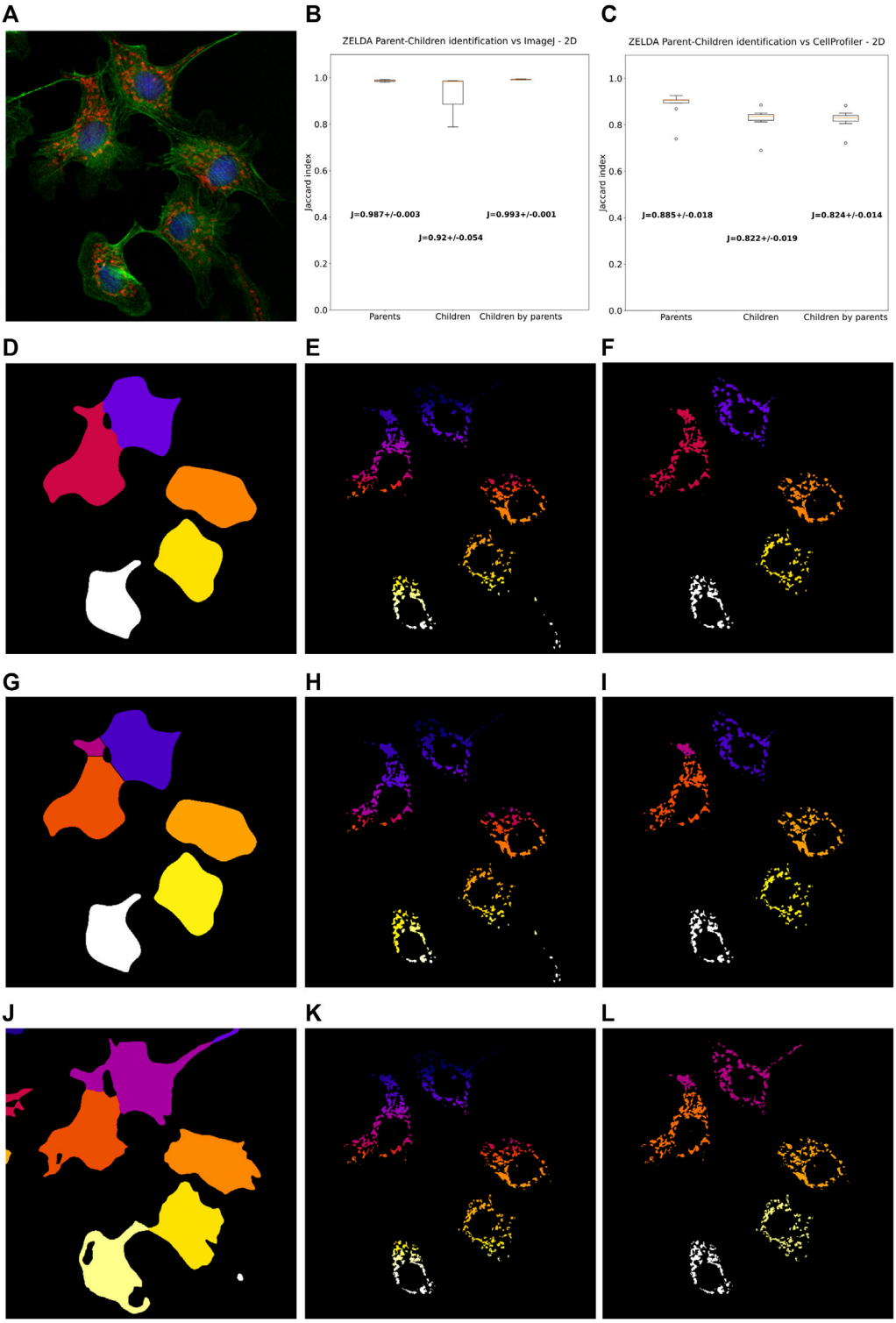


FIGURE 4 | ZELDA 2D segmentation and "parent-child" relation benchmarked with *ImageJ* and *CellProfiler*. **(A)** 2D images of BPAE cells stained with DAPI (blue, cell nuclei), AlexaFluor 488 (green, cytoplasms), and MitoTracker Red (red, mitochondria). **(B)** Labelling comparison between ZELDA and *ImageJ* showing an accordance above the 98% of the pixels for "parent" objects (cell cytoplasms), 92% for "child" objects (mitochondria), and 99% for the parent-child relation. **(C)** Comparison between ZELDA and *CellProfiler* showing a minor accordance still above the 88% of the pixels for "parents", 82% for "children", and 82% for the "parent-child" relation. ZELDA labelling of **(D)** cell cytoplasms, **(E)** mitochondria, and **(F)** masked mitochondria (parent-child relation). *ImageJ* labelling of **(G)** cell cytoplasms, **(H)** mitochondria, and **(I)** masked mitochondria (parent-child relation). *CellProfiler* labelling of **(J)** Cell cytoplasms, **(K)** mitochondria, and **(L)** masked mitochondria (parent-child relation).

Alternatively, the installation can be done downloading the repository, navigating to it with the *Anaconda* prompt and using the command "pip install -e." within the downloaded folder.

The plugin widgets have been created using *magicgui* (https://github.com/napari/magicgui), while the GUI plots included in the "Data Plotter" protocol are obtained with *matplotlib.backends.backend_qt5agg* (https://matplotlib.org/2.2.2/_modules/matplotlib/backends/backend_qt5agg.html).

The template for the plugin has been obtained from *cookiecutter-napari-plugin* (https://github.com/napari/cookiecutter-napari-plugin).

JSON Database for Modularity of the GUI and Customization of Image Analysis Protocols

Once the user has selected a specific base protocol, a JSON file is used by the plugin to load the right widgets in the GUI.

The "Design a new Protocol" option saves the custom workflow as a list of widgets that will be sequentially loaded the next time that the newly created protocol is launched. It will be visible just after re-launching *napari* and ZELDA plugin.

RESULTS

ZELDA Protocols as an Easy Way to Run Image Analysis Workflows for 2D and 3D Segmentation

ZELDA plugin for *napari* ("*napari-zelda*") makes available to the end-user the segmentation, measurement, and "parent-child" relation of two object populations. It ultimately allows to plot the results and explore the data in the same Graphical User Interface (GUI).

The current version of the plugin includes three different "protocols" to ease the image analysis of 3D data sets. Each protocol is a set of individual steps (functions) that return images (as *napari* layers), or results (printed plots in .tiff or tables in .csv format).

The first protocol, called "Segment a single population" (**Figure 1A**), can be used to segment both the 2D or 3D data sets. The basic workflow of this protocol (**Figure 1B**) includes simple steps, such as Gaussian Blur, Threshold, and Distance Map, to identify the seed points for the subsequent segmentation of the objects of interest. The user can then set the "min dist" parameter in the "Show seeds" function to improve the accuracy of cell counting, before calling the "Watershed" segmentation (**Figures 1C–H**). The detected objects can eventually be measured and the results table automatically saved (**Supplementary Figure S2A**).

Similar workflows have been previously implemented in useful tools such as *MorphoLibJ* (Legland et al., 2016) and in the latest versions of *CellProfiler* (McQuin et al., 2018), although with the limitation of being exclusively applied to the 2D image analysis, or lacking an embedded and flexible 3D viewer. In contrast, ZELDA provides an integration of a basic 3D object segmentation workflow with *napari* 3D rendering GUI. Notably, in ZELDA the individual workflow steps are also accessible as single functions that can be optionally used, or fine-tuned individually, without having to restart the entire workflow from scratch.

The second protocol, "Segment two populations and relate" (**Figure 2A**), implements the segmentation of two populations of objects in parallel, using the same workflow described above, with an additional step that allows establishing the "parent-child" relation between the two object populations (**Figures 2B–D**).

To run reproducible image analysis with ZELDA, both described protocols include a "log" functionality that stores the parameters used at each step. The log is shown in the GUI and can be optionally saved as a .txt file, together with the other results (**Supplementary Figure S2B**).

Once segmented, measured, and optionally related two object populations, the "Data Plotter" protocol (**Figure 2E**) allows to load a result table, and plot histograms or scatterplots of the measured properties. The plots are shown directly in the *napari* GUI and can be automatically saved as images to a specific folder. This has the advantage of avoiding the employment of additional software for data visualization.

Given that ZELDA does not require any coding skill, life science researchers are hugely facilitated by the integration of multiple bioinformatics tools in a single GUI.

Modularity of the ZELDA Graphical User Interface Allows to Easily Customize Bioimage Analysis Workflows Without Any Scripting Knowledge

Computer scientists and developers continuously propose new algorithms to tackle biological problems that frequently require extensive coding skills. However, users might have the necessity to reproduce a specific published workflow (such as the one in **Figure 1B**), without knowing a scripting language or necessarily having any background in image analysis. We made this possible by implementing a method that allows the customization of the image analysis protocols available in ZELDA. Indeed, by simply running the fourth option called "Design a New Protocol", a user can create a new custom protocol (**Figure 3A**). Every step of the base protocols is listed in a JSON database and the relative GUI widgets (used for the software layout) are available as ready-to-use modules to build personalized protocols. The different functions, such as threshold, gaussian blur or distance map etc., can be chosen in a drop-down menu at specific steps of the new protocol (**Figure 3B**). By using the saving option (**Figure 3C**), the JSON database will be automatically updated (**Figure 3D**), and the ordered series of GUI widgets will be available the next time that ZELDA plugin will be launched (**Figure 3E**). Once developed, custom protocols can be shared with the community using the "Import and Export Protocols" option (**Figure 3F**).

ZELDA Segmentation and Parent-Child Relation Have the Same Accuracy of *ImageJ* and *CellProfiler* in 2D and 3D Data Sets, and the Execution Is Twice Faster

In order to assess the accuracy in the segmentation of 2D and 3D data sets, we compared the results obtained by the ZELDA plugin

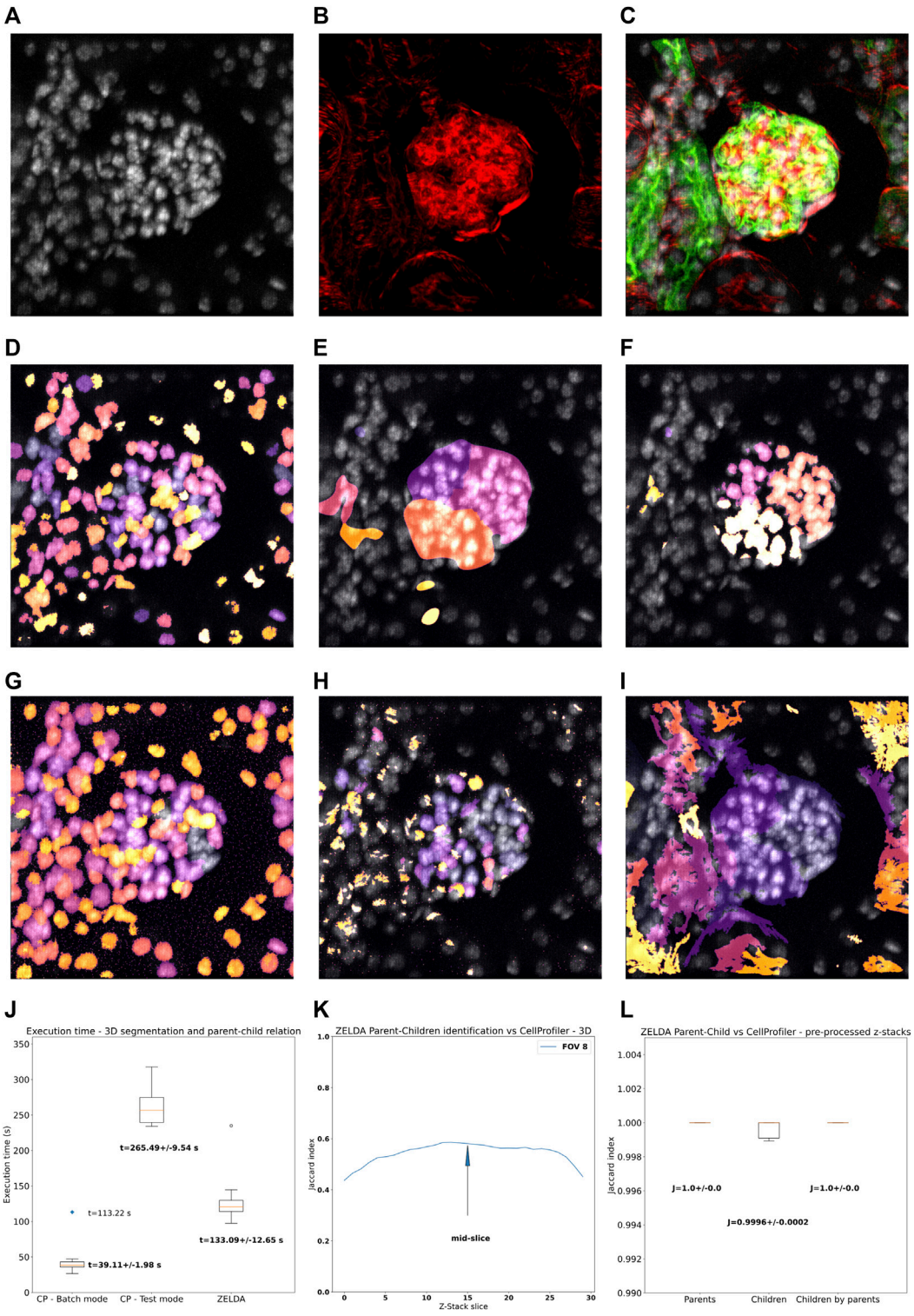


FIGURE 5 | 3D segmentation, "parent-child" relation, and execution time of ZELDA compared to *CellProfiler*. Z-stacks of mouse kidney tissue showing glomeruli were used for the benchmark in 3D. **(A)** DAPI staining used to segment the cell nuclei. **(B)** Phalloidin used to identify the glomerular structures **(C)** 3D rendering showing the merge of DAPI (gray), WGA (green), and phalloidin (red). **(D)** Nuclei, **(E)** glomerular structures, and **(F)** masked nuclei (parent-child relation) labelled by ZELDA in 3D. **(G)** Nuclei, **(H)** glomerular structures, and **(I)** masked nuclei (parent-child relation) labelled by *CellProfiler* in 3D (using a pipeline containing only 3D data compatible modules). **(J)** Execution time for the same workflow developed as both *CellProfiler* pipeline and ZELDA protocol, with the goal to segment in 3D and relate parents and children objects. The boxplots represent the distribution of multiple runs analyzing individual FOVs. For *CellProfiler* in batch mode, the CPU time has been considered,

(*Continued*)

FIGURE 5 | while the blue dot represents the total duration experienced by the end user for the analysis of 9 FOVs (including the wall time). **(K)** Variation of the Jaccard index of the segmentation obtained with ZELDA and *CellProfiler*, around the mid-slice where the signal is stronger. In the 3D case, the maxima of the Jaccard scores along the Z-stack were used for the benchmarking. Not all the *CellProfiler* modules are 3D compatible, then the execution of a minimal pipeline may result in over-segmented structures. The reason was identified to be the lack of a unique name for the same operation in 2D and 3D ("Smooth"), or the absence of 3D equivalents for some modules like the "ExpandOrShrink" morphological operations. *CellProfiler* might be able to process the data sets equivalently to ZELDA but with a longer and more complicated pipeline. **(L)** Increment of agreement on segmentation above the 99% once a pre-processed 3D data by ZELDA was proposed to *CellProfiler*, showing how quickly ZELDA can segment and relate in 3D using less steps than a *CellProfiler* pipeline.

for *napari* with those generated by two of the most widely used software in the bioimage field: *ImageJ* and *CellProfiler*.

As 2D data sets, we used images of cells (**Figure 4A**) at a low confluence (~30% of the field of view area) with a cytoplasmic staining to identify parent objects, and a second one for cellular organelles (children objects), with the final goal of correctly assign the organelles to the containing cell (parent-child relation).

Intriguingly, ZELDA performed almost equivalently to *ImageJ* (**Figure 4B**) in identifying parent objects (Jaccard index J = 0.987 ± 0.003), child objects (J = 0.920 ± 0.054), and in the parent-child relation (J = 0.993 ± 0.001). This means that, assuming *ImageJ* segmentation as ground truth (**Figures 4G–I**), ZELDA will correctly label the pixels of an organelle as belonging to the corresponding cell cytoplasm in 99% of the cases (**Figures 4D–F**).

However, the adherence with *CellProfiler* labelling was slightly less striking (**Figure 4C**) although this difference might be due to the many more parameters available in the *CellProfiler* GUI, such as the "declump method" in the "watershed" module etc., that haven't been implemented in ZELDA GUI to keep the software interface and its utilization as simple as possible. Nonetheless, the agreement on the identification of the parent cytoplasms found with *CellProfiler* (**Figure 4J**) was around 88% of the pixels, while for both the child objects segmentation and the parent-child relation (**Figure 4K–L**) it was ~82%.

Benchmarking the segmentation of 3D data sets has proven to be slightly more complicated, since not all the available modules in *CellProfiler* support the 3D data processing. For example, in version 4.2.1 the "smooth" module that operates a Gaussian blur filter, is available just for the 2D data pipeline, while another one has to be used for the 3D case. The same holds for morphological operations such as those executed by the "ExpandOrShrinkObjects". Trying to circumvent this lack of interchangeable 2D/3D functions could result in a more elaborated and time-consuming construction of the *CellProfiler* pipeline. Conversely, the versatile protocols supplied with ZELDA (**Figure 1**; **Figures 2A–D**) allowed the 3D segmentation and parent-child relation in fewer steps and about twice quicker than the *CellProfiler* "Test mode" (**Figure 5J**).

We then analyzed a collection of z-stacks of mouse kidney glomeruli, as 3D data sets (**Figures 5A–C**). In this tissue, phalloidin staining (**Figure 5B**) was used for the identification of the glomerular structures, and DAPI staining (**Figure 5A**) to pinpoint the cell nuclei contained in each glomerulus. The resulting segmentation of the two populations and parent-child relation obtained by ZELDA (**Figures 5D–F**) were compared with the output of a *CellProfiler* pipeline which included solely the 3D data compatible modules (**Figures 5G–I**).

Unfortunately, the labelling agreement between the two software was reduced with respect to the 2D analysis. A performance comparison of the 3D segmentation revealed a variation of the Jaccard index across the z-stack, with maximum values typically around the mid-slice, where the staining intensity of the confocal microscopy data set was stronger (**Supplementary Figure S3A**). We then considered the maxima of the Jaccard index across the z-stacks (**Figure 5K**), assessing an accordance around 63% for the parent objects (Jaccard index J = 0.632 ± 0.033), 73% for the children (J = 0.735 ± 0.032), and of 64% for the parent-child relation (J = 0.643 ± 0.029) (**Supplementary Figure S3B**).

We further investigated the reason for the lack of agreement on 3D data sets labelling between ZELDA and *CellProfiler*, and found that the difference was due to the absence of 3D equivalents for some modules (e.g., the "ExpandOrShrink" morphological operations), or lack of a unique naming for the 2D and 3D version of the same method in *CellProfiler* (e.g., "Gaussian Blur"). Indeed, pre-processing the z-stacks with the ZELDA and proposing the resulting smoothed 3D data sets to *CellProfiler*, successfully increased the accordance in identifying parents, children, and parent-child relation above the 99% of the pixels (**Figure 5L**).

Therefore, ZELDA can represent a faster interactive alternative to *CellProfiler* for the exploratory analysis of 3D data sets.

DISCUSSION

Many tools are available for 2D segmentation, while fewer are able to process 3D data sets (Schindelin et al., 2012) (McQuin et al., 2018) (Berg et al., 2019). The main limitation is frequently due to the lack of a flexible 3D viewer to render the resulting processed images (segmented volumes/surfaces) or visualize in an easy and understandable way the overlap between the labels assigned to each object and the original image. Additionally, many functions required for a complete 3D analysis workflow may demand different levels of background knowledge in coding and image analysis.

Considering the growing request for bioimage analysis tools and the difficulties encountered by the users, we developed ZELDA, a plugin for 3D image segmentation, and parent-child relation for microscopy image analysis in *napari* (napari contributors, 2019).

ZELDA plugin has the flexibility of being applicable to different purposes and data sets, such as the image measurement of beads to assess microscope resolution (**Figure 1B**), the RNA quantification in influenza-infected human cell nuclei (**Figures 2B–D**), the identification of

cellular compartments and organelle counting in cell culture samples (**Figures 4D–F**), or the morphological characterization of organs and tissues (**Figures 5D–F**).

The 2D and 3D image analysis workflows that ZELDA protocols convey (**Figure 1A**; **Figure 2A**) do not require an extensive knowledge of the used algorithms, coding skills, or an elevated number of "point and click" interactions.

The "Data Plotter" protocol (**Figure 2E**) enables the data exploration during the image analysis, favoring the biological sample comprehension, and potentially highlighting differences between treatments "on the fly". Furthermore, the reproducibility of workflows is sustained by the implementation of the log (**Supplementary Figure S2B**) and persistence in memory of the previously used image analysis parameters (i.e., restarting the same protocol will show the parameters values used during the last run).

The implementation of image analysis workflows found in literature is achievable with a fourth protocol called "Design a New Protocol" (**Figures 3A–C**). Without any scripting, users can manage the available "widgets" to create a custom GUI (**Figure 3E**) that can then be saved and shared with the community (by sharing the JSON database) (**Figure 3D**).

Nonetheless, through the customization of the GUI allowed by the fourth protocol, a simply different use of the already available functionalities can lead to better object segmentation. For example, including an additional "Threshold" step after the "Get DistanceMap", in a newly designed protocol, could help to remove smaller debris before "Show seeds". Certainly, the possibility of rearranging the components of the image analysis workflows, by using an immediate graphical mode, represents a valuable contribution as an open-source software to bioimage analysis.

To date, ZELDA presents a minimalist interface with three basic protocols implementing image analysis workflows, but it could be easily powered up with additional processing steps to improve image segmentation (e.g., morphological operators to moderate under and over-segmentation, a filter module to exclude segmented objects by intensity or shape descriptors, or allowing to deconvolve the data set before segmenting it).

Although still unable to process images in batch mode, ZELDA can find its niche of application as interactive software since we showed, by benchmarking, that it performs at a comparable level with *ImageJ* and *CellProfiler* in 2D. While in 3D, the segmentation and "parent-child" relation of multi-class objects is performed with a shorter implementation of the workflows and twice faster.

In conclusion, ZELDA plugin for *napari* can accelerate and facilitate the applications of bioimage analysis to life science research.

DATA AVAILABILITY STATEMENT

The data sets generated and analyzed in this study can be found in the GitHub repository https://github.com/RoccoDAnt/napari-zelda and on Zenodo doi: 10.5281/zenodo.5651284.

AUTHOR CONTRIBUTIONS

RD'A developed the *napari-zelda* plugin, acquired and analyzed the data, and wrote the manuscript. GP helped to develop the plugin and wrote the manuscript.

FUNDING

This work was supported by the Francis Crick Institute which receives its core funding from Cancer Research United Kingdom (FC001999), the United Kingdom Medical Research Council (FC001999), and the Wellcome Trust (FC001999).

ACKNOWLEDGMENTS

We thank Sara Barozzi (IEO, Milan) for the preparation of the PML NB sample shown in **Supplementary Figure S1A**. We thank Olivia Swann (Barclay Lab, Imperial College London) for supplying the influenza infected cells analyzed in **Supplementary Figure S1D**.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2021.796117/full#supplementary-material

REFERENCES

Berg, S, Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., et al. (2019). *ilastik: interactive machine Learn. (bio)image analysisNature Methods* 16, 1226–1232. doi:10.1038/s41592-019-0582-9

Fiorentino, A., Christophorou, A., Massa, F., Garbay, S., Chiral, M., Ramsing, M., et al. (2020). Developmental Renal Glomerular Defects at the Origin of Glomerulocystic Disease. *Cel Rep.* 33, 108304. doi:10.1016/j.celrep.2020.108304

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* 9, 90–95. doi:10.1109/MCSE.2007.55

Jamali, N., Dobson, E., Eliceiri, K., Carpenter, A., and Cimini, B. (2021). 2020 BioImage Analysis Survey: Community Experiences and Needs for the Future. *Biol. Imag.*, 1–28. doi:10.1017/S2633903X21000039

Lallemand-Breitenbach, V., and de Thé, H. (2018). PML Nuclear Bodies: from Architecture to Function. *Curr. Opin. Cel. Biol.* 52, 154–161. ISSN 0955-0674. doi:10.1016/j.ceb.2018.03.011

Legland, D., Arganda-Carreras, I., and Andrey, P. (2016). MorphoLibJ: Integrated Library and Plugins for Mathematical Morphology with ImageJ. *Bioinformatics* 32 (22), 3532–3534. doi:10.1093/bioinformatics/btw413

Long, J. S., Mistry, B., Haslam, S. M., and Barclay, W. S. (2019). Host and Viral Determinants of Influenza A Virus Species Specificity. *Nat. Rev. Microbiol.* 17, 67–81. doi:10.1038/s41579-018-0115-z

Martins, G. G., Cordelières, F. P., Colombelli, J., D'Antuono, R., Golani, O., Guiet, R., et al. (2021). Highlights from the 2016-2020 NEUBIAS Training Schools for Bioimage Analysts: a success story and Key Asset for Analysts and Life Scientists. *F1000Res* 10, 334. doi:10.12688/f1000research.25485.1

McKinney, W. (2011). *Pandas: A Foundational Python Library for Data Analysis and Statistics*.

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B. A., Karhohs, K. W., et al. (2018). CellProfiler 3.0: Next-Generation Image Processing for Biology. *Plos Biol.* 16 (7), e2005970. doi:10.1371/journal.pbio.2005970

Missiroli, S., Perrone, M., Genovese, I., Pinton, P., and Giorgi, C. (2020). Cancer Metabolism and Mitochondria: Finding Novel Mechanisms to Fight Tumours. *EBioMedicine* 59, 102943. doi:10.1016/j.ebiom.2020.102943

Miura, K., Paul-Gilloteaux, P., Tosi, S., and Colombelli, J. (2020). "Workflows and Components of Bioimage Analysis," in *Bioimage Data Analysis Workflows. Learning Materials in Biosciences*. Editors K. Miura and N. Sladoje (Cham: Springer). doi:10.1007/978-3-030-22386-1_1

napari contributors (2019). *Napari: A Multi-Dimensional Image Viewer for python*. doi:10.5281/zenodo.3555620

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., and Thirion, B. (2011). Scikit-learn: Machine Learning in Python. *J. Machine Learn. Res.* 12, 2825–2830. https://arxiv.org/abs/1201.0490v4.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an Open-Source Platform for Biological-Image Analysis. *Nat. Methods* 9, 676–682. doi:10.1038/nmeth.2019

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., et al. (2014). Scikit-Image: Image Processing in Python. *PeerJ* 2, e453. doi:10.7717/peerj.453

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 (3) 261–272. Epub 2020 Feb 3. Erratum in: Nat Methods. 2020 Feb 24;: PMID: 32015543; PMCID: PMC7056644. doi:10.1038/s41592-019-0686-2

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 Computer Science

BRIEF RESEARCH REPORT
published: 05 January 2022
doi: 10.3389/fcomp.2021.780026

An Open-Source Whole Slide Image Registration Workflow at Cellular Precision Using Fiji, QuPath and Elastix

Nicolas Chiaruttini[1], Olivier Burri[1], Peter Haub[2], Romain Guiet[1], Jessica Sordet-Dessimoz[3] and Arne Seitz[1]**

[1]Bioimaging and Optics Core Facility, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, [2]Imaging Consulting, Altlussheim, Germany, [3]Histology Core Facility, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Image analysis workflows for Histology increasingly require the correlation and combination of measurements across several whole slide images. Indeed, for multiplexing, as well as multimodal imaging, it is indispensable that the same sample is imaged multiple times, either through various systems for multimodal imaging, or using the same system but throughout rounds of sample manipulation (e.g. multiple staining sessions). In both cases slight deformations from one image to another are unavoidable, leading to an imperfect superimposition Redundant and thus a loss of accuracy making it difficult to link measurements, in particular at the cellular level. Using pre-existing software components and developing missing ones, we propose a user-friendly workflow which facilitates the nonlinear registration of whole slide images in order to reach sub-cellular resolution level. The set of whole slide images to register and analyze is at first defined as a QuPath project. Fiji is then used to open the QuPath project and perform the registrations. Each registration is automated by using an elastix backend, or semi-automated by using BigWarp in order to interactively correct the results of the automated registration. These transformations can then be retrieved in QuPath to transfer any regions of interest from an image to the corresponding registered images. In addition, the transformations can be applied in QuPath to produce on-the-fly transformed images that can be displayed on top of the reference image. Thus, relevant data can be combined and analyzed throughout all registered slides, facilitating the analysis of correlative results for multiplexed and multimodal imaging.

Keywords: ImageJ, Fiji, QuPath, WSI, registration, elastix, BigWarp, BigDataViewer

OPEN ACCESS

Edited by:
Florian Levet, UMR5297 Institut Interdisciplinaire de Neurosciences (IINS), France

Reviewed by:
Martin Leonard Jones, Francis Crick Institute, United Kingdom
John Bogovic, Janelia Research Campus, United States

****Correspondence:***
Nicolas Chiaruttini
nicolas.chiaruttini@epfl.ch
Arne Seitz
arne.seitz@epfl.ch

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *20 September 2021*
Accepted: *25 November 2021*
Published: *05 January 2022*

Citation:
Chiaruttini N, Burri O, Haub P, Guiet R, Sordet-Dessimoz J and Seitz A (2022) An Open-Source Whole Slide Image Registration Workflow at Cellular Precision Using Fiji, QuPath and Elastix. Front. Comput. Sci. 3:780026. doi: 10.3389/fcomp.2021.780026

INTRODUCTION

Whole slide imaging (WSI) is the process of acquiring images of tissue sections. It is typically realized in a line or a tile by tile scanning fashion in order to allow the imaging and visualization of samples exceeding the field of view of a conventional light microscope setup. Most of the commercially available systems accommodate at least two modalities: brightfield and fluorescence imaging. Within the last decade WSI turned out to become more and more important in pathology (Aeffner et al., 2019; Zarella et al., 2019) as well as in fundamental research.

Image processing of WSI images is challenging in many ways, in particular when aiming to register images from different modalities. Images often consist of gigapixel 2D planes, and matching correctly one slide to another usually requires a deformation more complex than a single affine transformation. Besides pure technical problems, life scientists face practical challenges if they want to use published methods. For instance, a grand challenge of non-linear image registration has been performed in 2018 (Borovec et al., 2020). However, it proved very difficult to apply any of the more successful methods, either because they were closed source or because the necessary documentation was not readily available. Moreover, very often a file conversion step is required in order to plug images in these workflows, and an extra effort is required to get the registration result in a usable way for downstream analysis. In short, the quality of the multi-modal registration algorithm is only one factor among many others influencing the adoption/popularity of an imaging analysis workflow.

Based on our group's experience as a microscopy core facility imaging, the following combination of requirements is until now not met for non-linear WSI registration: open-source, easy to install, capable to read and display multiresolution image files without conversion, capable to freely deform and interactively display deformed images, giving reasonably fast and accurate results, and compatible with a modular and powerful analysis platform.

The individual tasks and requirements mentioned above are however met in different pieces of software:

- For multiresolution file format reading, the OME consortium has built a library, Bio-Formats (Linkert et al., 2010), which can read more than a hundred microscopy oriented file formats. Some formats are unfortunately incompatible with Bio-Formats (3D HISTECH .mrxs) or are not multiresolution (Leica .lif). In these cases file conversion is unfortunately still required to be compatible with open source solutions. If one wants to use open source solutions for data analysis, the output file format should be carefully considered when deciding to buy new instruments.
- QuPath (Bankhead et al., 2017) is a very popular and powerful software for WSI analysis. It supports Bio-Formats multiresolution API, is user interactive, open-source, easy to install, and has powerful analysis capacities, furthermore customizable through scripting.
- Thanks to the ImgLib2 (Pietzsch et al., 2012) library and BigDataViewer (Pietzsch et al., 2015), the open-source ImageJ/Fiji (Schindelin et al., 2012) software can arbitrarily deform, slice and display big images. Among many other plugins of the BigDataViewer ecosystem, BigWarp (Bogovic et al., 2016) can be used to perform manual registration of multiresolution datasets.
- For automated multi-modal registration, many open source libraries exist. We decided to use the powerful and well-established elastix library (Klein et al., 2010; Shamonin, 2013). Elastix is well documented and provides flexibility in terms of registration possibilities, transformation models (affine, spline, etc.), optimizer settings and resolution.

In the following we present a fast and easy to use workflow for whole slide image registration. QuPath, a widely used open-source project for whole slide image display and analysis, serves as the entry point and can be used for further downstream analysis after the registration. This workflow is targeted towards life scientists lowering the entry barrier for whole slide registration and making it more user-friendly and reproducible.

RESULTS

As mentioned above, all the core open source elements needed to build a non-linear WSI registration workflow already exist, however, they do suffer from a lack of interoperability. For instance elastix is a C++ library with python wrappers, without pre-compiled Java wrappers. Moreover, while Fiji and QuPath are both Java-based software, they are incompatible in terms of Java versions. It is difficult however to find and allocate resources in order to make the connections between these components. Nevertheless, bridging these components is needed for life scientists, and this work is the result of such an effort. To bring these components together and provide a complete WSI non-linear registration workflow, the following missing components have been identified and implemented:

- A Fiji/BigDataViewer opener for QuPath projects
- Support of Bio-Formats multi-resolution API for BigDataViewer
- Fiji plugins to bridge elastix and Fiji
- A way to apply, store and reopen the deformation function which makes the correspondence between coordinates of registered images.

This work provides these missing pieces and thus closes the gap to a complete open-source workflow for non-linear WSI registration. It starts from the opening/display/definition of the images down to providing modular tools for the analysis in QuPath. An overview of the workflow is presented in **Figure 1**, as well as the locations of the missing components that were needed to make it work.

In short, all images which need to be registered need to be put into a single QuPath project, which is then opened in BigDataViewer. The reference image is used to align and transform all the other images. The non-reference images will be referred to as the moving images. Pairs of reference and moving images are then registered within Fiji, and transformations are exported to QuPath. The analysis and correlation between images can be performed in QuPath by transforming regions either towards the reference or the moving image. It is also possible to generate an image combining the reference and registered images into a new QuPath entry.

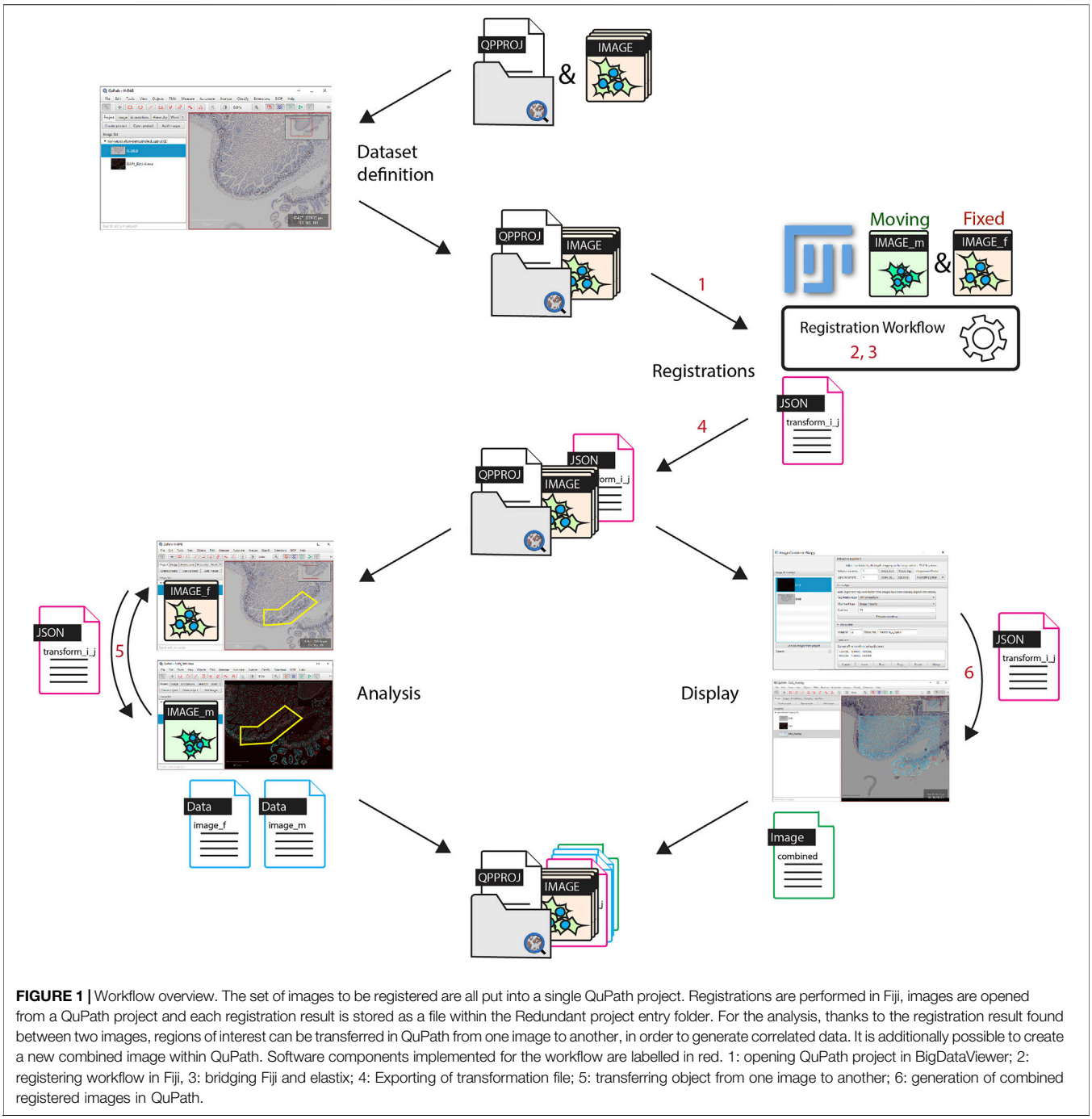


FIGURE 1 | Workflow overview. The set of images to be registered are all put into a single QuPath project. Registrations are performed in Fiji, images are opened from a QuPath project and each registration result is stored as a file within the Redundant project entry folder. For the analysis, thanks to the registration result found between two images, regions of interest can be transferred in QuPath from one image to another, in order to generate correlated data. It is additionally possible to create a new combined image within QuPath. Software components implemented for the workflow are labelled in red. 1: opening QuPath project in BigDataViewer; 2: registering workflow in Fiji, 3: bridging Fiji and elastix; 4: Exporting of transformation file; 5: transferring object from one image to another; 6: generation of combined registered images in QuPath.

The registration workflow, which happens within Fiji is described in **Figures 2A,B**. In brief: for each pair of images that needs to be registered, the user has to choose the reference image and then select among 4 (optional) successive registration steps: rigid manual B.1—affine auto B.2—spline auto B.3 (based on landmarks)—spline manual B.4.

Step 1. is a manual rigid registration step which is required if the images are not approximately aligned initially. This may happen because Bio-Formats uses the stage location of the microscope to position the images in physical space. If the images to register originate from the same microscope, keeping this offset may be helpful. In other cases, there's a chance that images won't overlap at all. The Bio-Formats induced offset can thus be removed at the start of the workflow, and the user can furthermore rigidly move the moving image to improve the alignment before resuming the next steps.

Step 2. consists of an automated affine registration with elastix (see elastix parameters in **Supplementary Appendix 1**). Since the

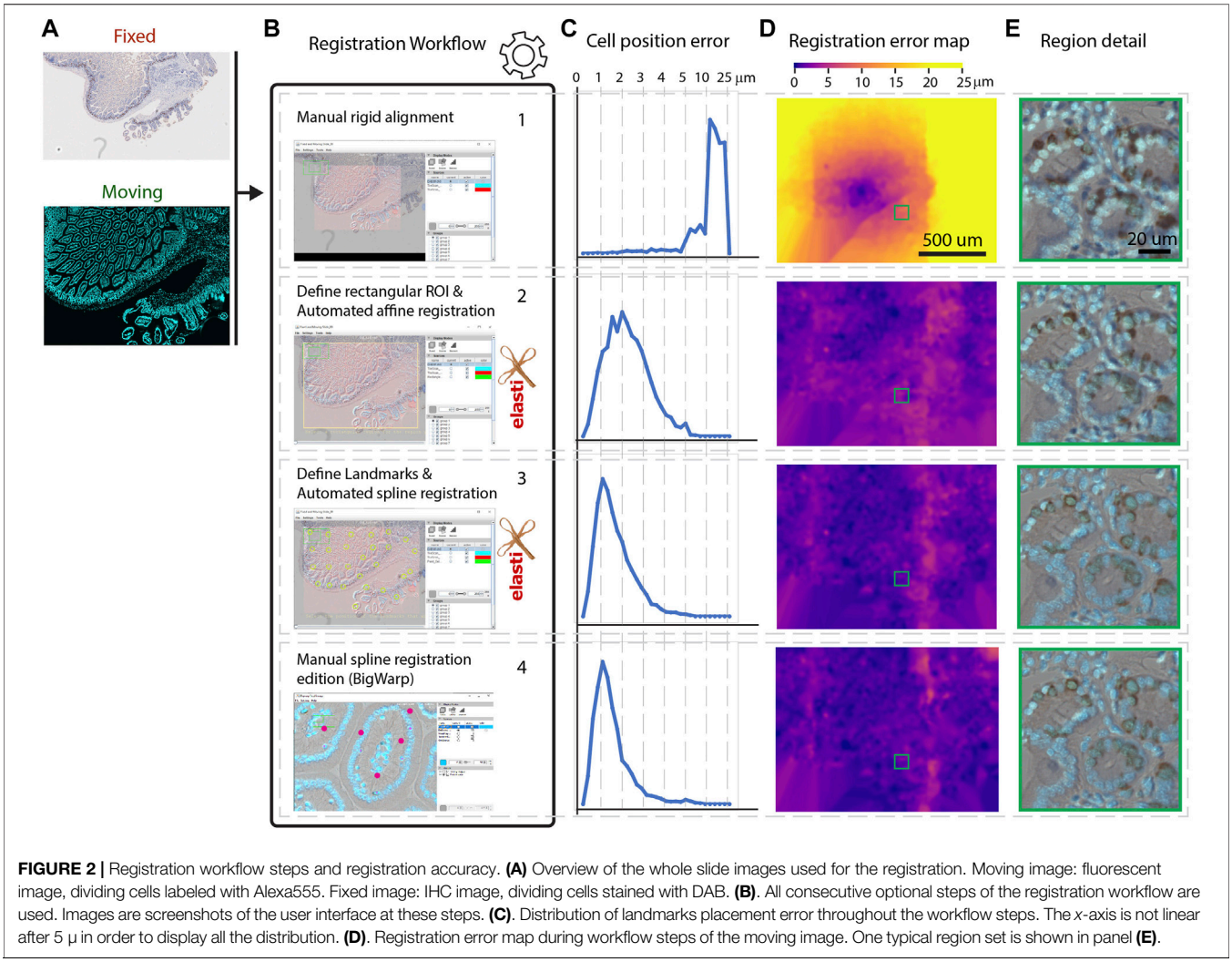


FIGURE 2 | Registration workflow steps and registration accuracy. **(A)** Overview of the whole slide images used for the registration. Moving image: fluorescent image, dividing cells labeled with Alexa555. Fixed image: IHC image, dividing cells stained with DAB. **(B)**. All consecutive optional steps of the registration workflow are used. Images are screenshots of the user interface at these steps. **(C)**. Distribution of landmarks placement error throughout the workflow steps. The *x*-axis is not linear after 5 µ in order to display all the distribution. **(D)**. Registration error map during workflow steps of the moving image. One typical region set is shown in panel **(E)**.

images are expected to be very large, this affine registration is performed on images that are first resampled at 10 µm per pixel within a rectangular region of the physical space. The moving and the fixed image may cover different physical areas, as in **Figure 2A**. In the user interface, a rectangular region covering entirely both images is first suggested to the user. This first guess can be modified interactively and thus modified if necessary.

Step 3. consists of a semi automated spline registration. For this step, a set of user-defined patches are each registered automatically in an affine manner (using the same elastix parameters as step 2). With the default parameters, each patch is a 500 × 500 µm area resampled at 1 µm per pixel. This xy sampling is typically sufficient for images from histological slides where objectives often have a numerical aperture below 1.0. For an accurate registration, these patches should be content-rich, and approximately correctly aligned, a condition met if the first coarse registration is successful. To extend the patches registration over the whole image, only the location of the central point (moving/fixed) of each patch is kept to build a landmark based registration. These points are used to extrapolate the transformation over the whole slide using B-Splines of the ImgLib2 library, as in the BigWarp plugin (Bogovic et al., 2016). We therefore assume a continuity in the deformation field beyond the user-defined patches. This has two advantages: 1—only sub-regions of the images have to be loaded, increasing the speed of the workflow; 2—if the user uses well defined sparse regions, the registration will be robust to missing tissue regions. Performing sparse registrations however has a disadvantage: a discontinuous deformation (for instance with an object split in two shifted rigid parts) will be ill registered.

Step 4. consists of manual spline correction. This step calls the BigWarp plugin (Bogovic et al., 2016) with pre-registered images from the previous steps. If an automated spline registration has been performed (Step 3), BigWarp is initialized with the registered landmarks positioned from the previous step. This allows the user to correct the output of the automated spline, as well as to add new landmarks to be more precise in some particular regions, if needed. Providing a way to manually edit

the result of an automated workflow can save a significant amount of time when the automation quality is not sufficient for a particular task. This BigWarp manual registration part can be repeated many times in case several rounds of manual corrections are needed. In this case, the previous registration is loaded into BigWarp as the initial state which can be modified and resaved.

At the end of the workflow, a transformation file defining how coordinates can be converted from the moving image to the reference image is stored within the QuPath reference image entry folder.

In order to demonstrate the performance of this workflow, we applied these four consecutive steps on a test dataset. It consists of a section of a fixed mouse duodenum. Dividing cells have been stained consecutively with two different methods. The fluorescence based 5-Ethynyl-2′-deoxyuridine (EdU) staining with click chemistry was followed by a 3,3′-Diaminobenzidine (DAB) revealing using Anti-Bromodeoxyuridine (BrdU) antibodies, leading to the immunohistochemistry (IHC) image. Each staining step required manipulation of the sample (see materials and methods). This sample requires alignment and registration because it consists of two multi modal images (one fluorescent, one IHC) where different zones of the sample were imaged, with different cameras, and some stitching artefacts are present. However, note that the exact same cells are imaged in the different modalities, which is a requirement for this workflow (i.e. it has not been tested for serial sections). Since the same (dividing) cells are targeted in both images, we expect a very good match between the positively detected cells in both modalities.

To assess the quality of the registration in the different steps of the workflow we compare against a ground-truth registration using BigWarp by identifying and placing more than 3,000 landmarks manually, covering the whole slide on identifiable features, in most of the cases individual cells. Making the ground truth required around 6 h of work. Note that on average the uncertainties on the placement of landmarks for the ground truth is of about 1 µm, leading to some uncertainties even in the deformation ground truth.

We then assessed the quality of the registration at each step of the workflow, by computing the distance of these landmarks to their target location in the ground truth (**Figure 2C**.). **Figure 2D** shows errors map throughout the steps: how far each landmark is to its ground truth is color coded and displayed in the coordinates of the moving image. A typical zoomed in region is displayed in **Figure 2E** for all successive steps.

In step 1, we can see that a 10 µm shift is present and this is clearly not precise enough for individual cells to match. Conversely, as in the ground truth, it is hard to distinguish meaningful differences when landmarks are less than 2 µm from their ground truth location.

Based on **Figures 2C–E**, we can conclude that an automated affine registration already brings good global positioning, since the vast majority of landmarks are positioned within 5 µm of error. The automated spline registration, which has been performed with 32 manually placed landmarks, greatly improves the quality of the registration, since 90% of the cells now have an error below 3 µm. Spending additional time to manually fix registration mismatch in some regions improves the overall registration quality further (92% of the cells have an error below 3 µm). In terms of timing, the automated part of the workflow (steps 2 and 3) takes around 2 min on a 4 core entry-level laptop. 5 additional minutes were needed to perform the corrections during step 4.

Figure 3 summarizes how the registration output can be easily plugged into QuPath in order to further analyze the registered images. We first annotated a region hosting most of the dividing cells in the fluorescent (moving) image. We segmented all cells (4,393) within this region using the DAPI channel of the fluorescent image (moving) by using Stardist in its default versatile 2D fluorescent model (Schmidt et al., 2018; Weigert et al., 2020). Thanks to the generated transformation file, the annotations and detections can be transformed and transferred to their target location in the (fixed) DAB image (**Figure 3A**). Cell properties can then be measured and compared between both images. We choose to segment cells in the fluorescence channel since it is easier to discriminate cells in the DAPI modality. We are thus transferring cell regions of interest (ROIs) from the moving image to the fixed image. Note that the deformation computed in this workflow is invertible, meaning that it is also possible to segment cells regions in the fixed image and then transfer them to the moving image.

For downstream analysis in QuPath, we decided to provide a way to transform a region of interest from one slide to another. Since regions of interest are defined with double precision coordinates, very little precision is lost when transforming coordinates. Conversely, generating a transformed image in QuPath creates some interpolation artefacts due to the pixel-based resampling. However, it may be useful to display transformed images over the reference image, e.g. for display and presentation purposes. This can be done with an additional QuPath extension provided in this workflow, called Image Combiner Warpy. With this tool, the transformed images can be overlaid on top of the reference image, based on the transform files and by using various possible sampling methods. Since this overlay process creates only one additional QuPath project entry, using the existing image data, it is executed almost immediately, with no delay time. If effects due to sampling artifacts do not matter, the combined image can be used for analysis in QuPath, including the transformed channels as well as the reference channel. If computation time and storage space are not an issue, and availability as a whole slide image file is important, the combined image project can be fully computed and resaved from within QuPath, for example as an OME tiff.

In **Figure 3B**, we assessed how the correlative measurements performed on cells are varying when each successive step of the workflow is applied (using transferred ROIs). We expect most DAB positive cells to be also positive for the fluorescent EdU stain, since both staining are for dividing cells. If each cell region is correctly transferred, we expect many double positive cells (EdU+/DAB+). To decide whether a cell is positive for DAB, we detected cells in the DAB image and found that a threshold of 0.18 OD in DAB is accurately discriminating dividing cells from non-dividing ones. We proceeded similarly for the fluorescence image

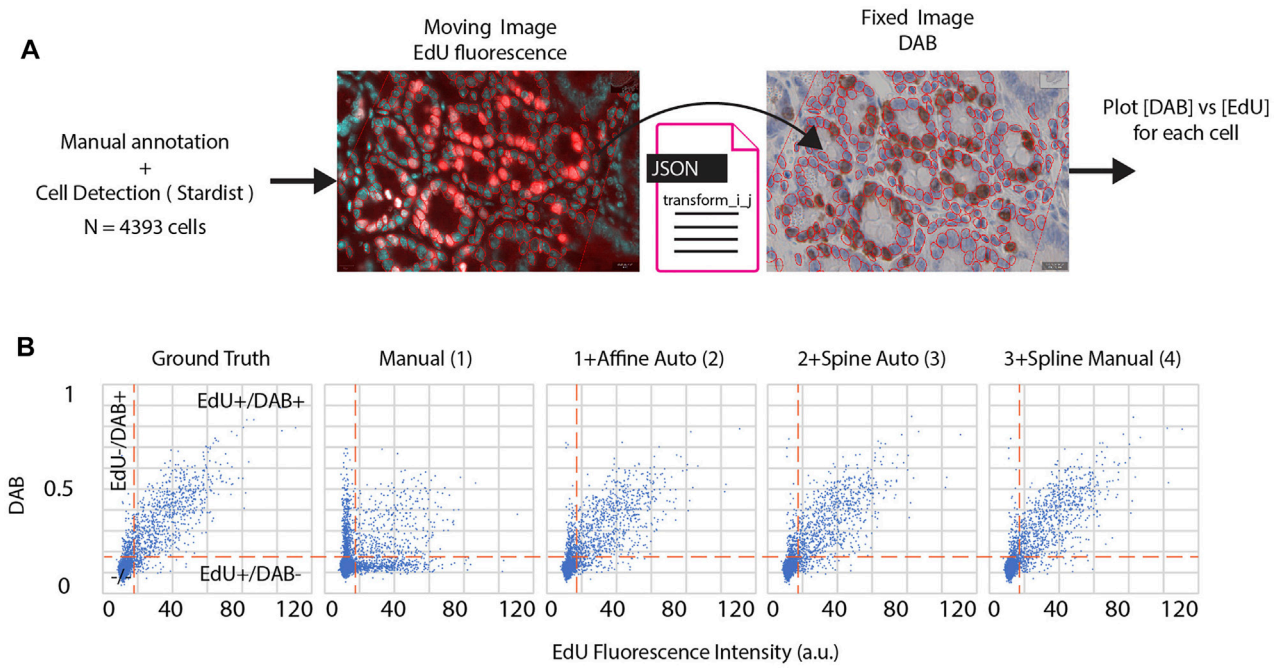


FIGURE 3 | (A) The EdU fluorescence image (moving image) is annotated and cells are detected thanks to the DAPI stain within a region of interest. Cells detected in the fluorescence image are transferred to the DAB image (fixed image), using a specific registration result file. Since the transformation is invertible, it would have been possible to segment cells in the DAB image and transfer cells to the fluorescent image.**(B)**. Each cell property for each image can be measured and combined during downstream analysis. Scatter plots of 4,393 detected cells, X: mean fluorescence EdU intensity, Y: mean DAB optical density. Each plot corresponds to a specific transform file, respectively from left to right: - manual annotation of 3,000 + landmarks (ground truth) - manual rigid transformation 1)—manual rigid transformation + automated affine transformation 2)—all steps in 2) + automated spline transformation 3)—all steps in 3) + manual spline transformation 4). Vertical dashed lines: fluorescence threshold (18 a.u.) separating EdU + cells from EdU- cells. Horizontal dashed lines: DAB OD threshold (0.18) separating DAB + cells from DAB- cells.

TABLE 1 | Comparison of cell classes depending on registration procedure. Counting of cell DAB/EdU classes (+/+, –/–, +/–, –/+) for the ground truth and for the different registration methods used (see **Figure 3**). In parenthesis is shown the excess or deficit of cells being counted when compared to ground truth. A lower number in absolute value means a result being closer to the one obtained with the ground truth deformation field.

		DAB+ / EdU+	DAB- / EdU-	DAB+/EdU-	DAB-/EdU+
	Ground Truth	790	3,353	160	90
1	Manual Rigid	386 (–404)	3,020 (-333)	493 (+333)	494 (+404)
2	(1) + Auto Affine	757 (–33)	3,279 (-74)	234 (+74)	123 (+33)
3	(2) + Auto Spline	767 (–23)	3,332 (-21)	181 (+21)	113 (+23)
4	(3) + Manual Spline	778 (–12)	3,341 (-12)	172 (+12)	102 (+12)

and set a threshold of 18 for the mean fluorescence over each detected cell region. If cell regions are incorrectly transferred, very little correlation between stains should be observed. Throughout each step of the workflow we then plot the mean DAB optical density versus the mean fluorescence EdU intensity per individual cell. These graphs allow counting cells which are considered to be dividing cells for both modalities (EdU+/DAB+), non-dividing in both modalities (EdU–/DAB–), and which give different outcomes based on the modalities (EdU+/DAB–) and (EdU–/DAB+).

We assess the distance of a registration to the ground truth by counting the number of cells which differ in this count (**Table 1**). In agreement with the quality of the registration shown in **Figure 2**, adding more registration steps reduces the difference to the ground truth. Relying only on a manual rigid registration is unsuitable for quantification in this case. The last step in the workflow (spline manual registration) is only giving a minor improvement (changing the classification of only 10 cells among 4,300) and might be omitted in some cases.

In summary the comparison of the DAB with the fluorescence images demonstrate that this workflow is able to reach high quality registration. Only 1.5% (12 out of 790) of the double positive cells were missed due to alignment errors.

DISCUSSION AND PERSPECTIVES

This workflow has been tested for the applications achieving multiplexing by imaging the same sample through several rounds

of staining. As the physical size of the image and the target resolution are known, reasonable default parameters for elastix can be used providing a user interface which is almost parameter free. This is particularly important as parameter fine tuning is cumbersome and challenging for non-experts. Omitting this step clearly lowers the entry barrier for the broader life-science community.

Until now this workflow has not been tested for the registration of serial sections where cellular matching from one slice to the next cannot be expected. Conversely in its present form this workflow is not of sufficient precision to match small sub cellular structures, such as membranous organelles. However, due to its modular design, this workflow can be adapted either by using different alignment parameters or by using other alignment methods, opening the way to support other use cases. In particular, elastix is the library we chose for registrations, but other libraries exist, including ones which are only available in python. By using PyImageJ (Rueden et al., 2021), we should be able to integrate cutting edge registration methods into this workflow.

CONCLUSION

We implemented a workflow for semi-automated non-linear WSI registration which requires minimal user input and can be realized in a few minutes per slide. It facilitates the registration of images for all applications where multiplexing is achieved via several rounds of staining. The performance is almost identical to that of a registration workflow based on manual annotation, but requires far fewer landmarks with the advantages of omitting the time consuming and tedious task of finding landmarks. Therefore it can save several hours of human annotation workload. Thanks to its modular design, it will be extended in the future to support other registration methods and other use cases, such as serial sections alignment (between them or with an atlas) and CLEM.

MATERIALS AND METHODS

Sample used for the demo registration (https://doi.org/10.5281/zenodo.5674521 : Mouse duodenum fixed in 4% PFA overnight at 4°C, processed for paraffin infiltration using a standard histology procedure and cut at 4 microns were dewaxed, rehydrated, permeabilized with 0.5% Triton X-100 in PBS 1x and stained with Azide - Alexa Fluor 555 (Thermo Fisher) to detect EdU and DAPI for nuclei. The images were taken using a Leica DM5500 microscope with a PL Fluotar 40X N.A.1.0 oil objective (grayscale CCD camera: DFC350FXR2, pixel dimension: 0.161 µm). Next, the slide was unmounted and stained using the fully automated Ventana Discovery xT autostainer (Roche Diagnostics, Rotkreuz, Switzerland). All steps were performed on automate with Ventana solutions. Sections were pretreated with heat using the CC1 solution under mild conditions. The primary rat anti BrDU (clone: BU1/75 (ICR1), Serotec, diluted 1:300) was incubated 1 h at 37°C. After incubation with a donkey anti rat biotin diluted 1:200 (Jackson ImmunoResearch Laboratories), chromogenic revelation was performed with DabMap kit. The section was counterstained with Harris hematoxylin (J.T. Baker) before a second round of imaging on DM5500 PL Fluotar 40X N. A.1.0 oil (RGB camera: DFC 320 R2, pixel dimension: 0.1725 µ). Before acquisition, a white-balance as well as a shading correction is performed according to Leica LAS software wizard. The fluorescence and DAB images were converted in ome.tiff multiresolution file with the kheops Fiji Plugin (Guiet et al., 2021).

All the tutorials (written and video) necessary to apply the workflow in practice are accessible in https://c4science.ch/w/warpy/ .

DATA AVAILABILITY STATEMENT

The dataset presented in this study can be found in an online repository: https://doi.org/10.5281/zenodo.5674521. Tutorials and links to the code repositories can be found in the documentation at https://c4science.ch/w/warpy/.

ETHICS STATEMENT

The studies involving animals were reviewed and approved by the cantonal veterinary authority (authorization number VD1890.2).

AUTHOR CONTRIBUTIONS

RG and JS-D designed and RG imaged the demo sample. NC and OB wrote the Warpy code. PH wrote the Image Combiner Warpy code. NC performed the data analysis. NC, PH, AS and RG wrote the manuscript.

ACKNOWLEDGMENTS

The histology data was prepared at the EPFL Histology Core Facility. We thank Nathalie Müller and Gian-Filippo Mancini for the preparation of the samples. We thank the Fiji and QuPath open source community for the continuous support they offer, especially Pete Bankhead, John Bogovic, Christian Tischer, Jan Eglinger and Curtis Rueden.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2021.780026/full#supplementary-material

REFERENCES

Aeffner, F., Zarella, M., Buchbinder, N., Bui, M., Goodman, M., Hartman, D., et al. (2019). Introduction to Digital Image Analysis in Whole-Slide Imaging: A white Paper from the Digital Pathology Association. *J. Pathol. Inform.* 10, 9. doi:10.4103/jpi.jpi_82_18

Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., et al. (2017). QuPath: Open Source Software for Digital Pathology Image Analysis. *Sci. Rep.* 7, 16878. doi:10.1038/s41598-017-17204-5

Bogovic, J. A., Hanslovsky, P., Wong, A., and Saalfeld, S. (2016). "Robust Registration of Calcium Images by Learned Contrast Synthesis," in 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), Prague, Czech Republic, 13-16 April 2016, 1123–1126. doi:10.1109/ISBI.2016.7493463

Borovec, J., Kybic, J., Arganda-Carreras, I., Sorokin, D. V., Bueno, G., Khvostikov, A. V., et al. (2020). ANHIR: Automatic Non-rigid Histological Image Registration Challenge. *IEEE Trans. Med. Imaging* 39, 3042–3052. doi:10.1109/TMI.2020.2986331

Guiet, R., Chiaruttini, N., Eglinger, J., and Burri, O. (2021). *BIOP/ijp-kheops: Zenodo Release*. doi:10.5281/ZENODO.5256256

Klein, S., Staring, M., Murphy, K., Viergever, M. A., and Pluim, J. (2010). Elastix: A Toolbox for Intensity-Based Medical Image Registration. *IEEE Trans. Med. Imaging* 29, 196–205. doi:10.1109/TMI.2009.2035616

Linkert, M., Rueden, C. T., Allan, C., Burel, J.-M., Moore, W., Patterson, A., et al. (2010). Metadata Matters: Access to Image Data in the Real World. *J. Cel Biol.* 189, 777–782. doi:10.1083/jcb.201004104

Pietzsch, T., Preibisch, S., Tomančák, P., and Saalfeld, S. (2012). ImgLib2-generic Image Processing in Java. *Bioinformatics* 28, 3009–3011. doi:10.1093/bioinformatics/bts543

Pietzsch, T., Saalfeld, S., Preibisch, S., and Tomancak, P. (2015). BigDataViewer: Visualization and Processing for Large Image Data Sets. *Nat. Methods* 12, 481–483. doi:10.1038/nmeth.3392

Rueden, C., Evans, E., Yang, L., Pinkert, M., Liu, Y., Hiner, M., et al. (2021). *Imagej/Pyimagej: v1.0.2 Zenodo*. doi:10.5281/ZENODO.5537065

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an Open-Source Platform for Biological-Image Analysis. *Nat. Methods* 9, 676–682. doi:10.1038/nmeth.2019

Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. (2018). "Cell Detection with Star-Convex Polygons," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018 Lecture Notes in Computer Science*. Editors A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger (Cham: Springer International Publishing), 265–273. doi:10.1007/978-3-030-00934-2_30

Shamonin, D. (2013). Fast Parallel Image Registration on CPU and GPU for Diagnostic Classification of Alzheimer's Disease. *Front. Neuroinform.* 7. doi:10.3389/fninf.2013.00050

Weigert, M., Schmidt, U., Haase, R., Sugawara, K., and Myers, G. (2020). "Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy," in 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1-5 March 2020, 3655–3662. doi:10.1109/WACV45572.2020.9093435

Zarella, M. D., Bowman;, D., Aeffner, F., Farahani, N., Xthona;, A., Absar, S. F., et al. (2019). A Practical Guide to Whole Slide Imaging: A White Paper from the Digital Pathology Association. *Arch. Pathol. Lab. Med.* 143, 222–234. doi:10.5858/arpa.2018-0343-RA

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 m ay be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers in Computer Science

METHODS
published: 04 February 2022
doi: 10.3389/fcomp.2021.769338

Automated Video Monitoring of Unmarked and Marked Honey Bees at the Hive Entrance

Iván F. Rodriguez[1,2], *Jeffrey Chan*[1,3], *Manuel Alvarez Rios*[3], *Kristin Branson*[4], *José L. Agosto-Rivera*[5], *Tugrul Giray*[5] *and Rémi Mégret*[3]*

[1] Department of Mathematics, University of Puerto Rico, Río Piedras Campus, San Juan, Puerto Rico, [2] Department of Cognitive, Linguistic and Psychological Sciences, Brown University, Providence, RI, United States, [3] Department of Computer Sciences, University of Puerto Rico, Río Piedras Campus, San Juan, Puerto Rico, [4] Howard Hughes Medical Institute, Janelia Research Campus, Ashburn, VA, United States, [5] Department of Biology, University of Puerto Rico, Río Piedras Campus, San Juan, Puerto Rico

We present a novel system for the automatic video monitoring of honey bee foraging activity at the hive entrance. This monitoring system is built upon convolutional neural networks that perform multiple animal pose estimation without the need for marking. This precise detection of honey bee body parts is a key element of the system to provide detection of entrance and exit events at the entrance of the hive including accurate pollen detection. A detailed evaluation of the quality of the detection and a study of the effect of the parameters are presented. The complete system also integrates identification of barcode marked bees, which enables the monitoring at both aggregate and individual levels. The results obtained on multiple days of video recordings show the applicability of the approach for large-scale deployment. This is an important step forward for the understanding of complex behaviors exhibited by honey bees and the automatic assessment of colony health.

Keywords: honey bee monitoring, bee counting, pollen detection, pose estimation, convolutional neural networks

OPEN ACCESS

Edited by:
Florian Levet,
UMR5297 Institut Interdisciplinaire de Neurosciences (IINS), France

Reviewed by:
Konrad Schindler,
ETH Zürich, Switzerland
Varun Kapoor,
Institut Curie, France

***Correspondence:**
Rémi Mégret
remi.megret@upr.edu

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *10 September 2021*
Accepted: *27 December 2021*
Published: *04 February 2022*

Citation:
Rodriguez IF, Chan J, Alvarez Rios M, Branson K, Agosto-Rivera JL, Giray T and Mégret R (2022) Automated Video Monitoring of Unmarked and Marked Honey Bees at the Hive Entrance. Front. Comput. Sci. 3:769338. doi: 10.3389/fcomp.2021.769338

1. INTRODUCTION

There is a growing interest in the quantification of behavior in honey bees (*Apis Melifera*). One of the major concerns is the dramatic yearly decrease in honey bee populations (Neumann and Carreck, 2010; Anderson et al., 2011; Huang and Giray, 2012), which has impacted the agriculture industry in the last few years. Although there are existing hypotheses regarding the usage of pesticides and urban growth, there are still questions unanswered that require precise observations to guide possible actions. Furthermore, honey bee colonies exhibit complex self-regulatory behaviors that are not yet fully understood. This includes how colonies maintain homeostasis or adapt to environmental changes, automatic adjustment of circadian patterns based on thermal cycles (Giannoni-Guzmán et al., 2021), thermo-regulation (Kaspar et al., 2018), or the individual variation in foraging activities in function of the season of the year (Meikle and Holst, 2016). Such studies may benefit greatly from automatic surveillance systems of the hives to detect both individual and collective behavior continuously over days or even seasons to provide crucial insights about biological mechanisms that express themselves over such time frames.

Foraging behavior has traditionally been studied through visual inspection of bees marked with number tags (Wario et al., 2015), which is a very time-consuming and error-prone process, as

foraging bees usually spend very little time in the ramp before entering the hive with their payload. The overall count of entrance and exits can currently be estimated using optical or mechanical bee counting sensors that detect the passage and direction of individual bees through gates at the entrance of the hive (Meikle and Holst, 2016). In that setup, only aggregated statistics can be obtained, which led to the development of sensor technology to detect entrance and exits of individuals marked with Radio-Frequency Identification tags (RFID) (de Souza et al., 2018).

A complementary approach is to analyze the behavior through video recordings. The advantage of the video collection at the colony entrance is that in addition to entrance and exit events, a much larger set of information is available: payload (pollen), the identity of bees with markings, morphology and phenotype, social interactions, gait, etc. This technique is minimally invasive since it does not require manipulation of the individuals, except to perform optional marking. One of the main challenges of this approach is the need for algorithmic processing of the videos which is not needed in simpler sensor setups. The application of machine vision to such applications has become in the last decade a very compelling option due to the improvement in computational power and artificial intelligence methods (Branson et al., 2009). The idea is to develop algorithms that can identify and quantify the behaviors of interest with limited human intervention. This enables the design of high throughput systems that can analyze long periods of video recordings automatically.

Ideally, the observation should happen in the most natural setup to avoid interfering with the behavior; however, there is a known trade-off between the complexity of the problem and the conditions of observation (Robie et al., 2017). As the set-up becomes as unobtrusive as possible, the complexity of the task rises, which requires more powerful algorithmic approaches to address it. This is the case at a normal entrance of a colony, where there are natural changes of illumination, changing the number of individuals, and fast moving individuals that make the problem more challenging. In addition, the entrance ramp can be very crowded at certain times of the day, which makes it difficult for traditional background subtraction based algorithms (Campbell et al., 2008) to detect precisely each individual.

In this study, we introduce a new computer vision system that leverages recent advances in deep learning to monitor collective and individual behavior of honey bees at the entrance of their hive with minimal interference. The rest of this study is organized as follows. In Section 2, an overview of related study is presented. The proposed system is presented in Section 3 and evaluated in Section 4. The main findings and future directions are discussed in Section 5.

2. RELATED WORK

We first review related studies about the monitoring of honey bees in the context of foraging analysis, then discuss more general methodologies for detection, tracking, and characterization of individual behavior from video.

2.1. Bee Counters and Forager Traffic Analysis

Bee counting has previously been addressed with the help of optical counters, RFID techology, and Machine Vision Counters. Refer to Meikle and Holst (2016) for a detailed review of existing approaches.

Optical counters consist of small tunnels that force the bees to pass through them one by one where they cross an optical beam that triggers a detection. The only crossing is detected, which limits the information to aggregated counting. Multiple commercial products exist for this task, such as Hive Mind Strength monitor, Arnia Remote Hive Monitoring, or Lowland Electronics's BeeSCAN and ApiSCAN-Plus, as well as open-source projects such as https://github.com/hydronics2/2019-easy-bee-counter.

Radio Frequency Identification (RFID) counters (de Souza et al., 2018) solve the identification problem by using individual lightweight radio frequency tags that are placed on the torso of the bees. Radio frequency detectors detect the entrance or exit of each tag that is associated with a unique ID. Due to the limited range of detection, bees are typically guided into tunnels. Detection is limited to tagged bees and requires the gluing of the tags on each individual to be monitored.

Machine vision based counters have been explored for about a decade now. These counters have a simple setup in terms of data capture (video camera at the entrance) and leave the complexity of the problem to algorithm development. Among the first ones to propose such a system, Campbell et al. (2008) detected bees using background subtraction and modeled the possible motion in subsequent frames to perform tracking. However, as they noted, a more detailed model of the bee orientation is necessary to avoid merging tracks when honey bees interact closely at the entrance. Similar detection and tracking approach was applied to flight trajectories in front of the hive by Magnier et al. (2018). Using stereovision sensors, Chiron et al. (2013) obtained 3D flight trajectories. Recent study explored approximate forager traffic quantization from the detection of motion (Kulyukin and Mukherjee, 2019).

More advanced detectors have incorporated the usage of tags (Chen et al., 2012; Boenisch et al., 2018). These tags contain 2d barcodes that can be detected by computer vision algorithms. Such detection can be used for both the detection of individual entrance/exit and identification. Compared to RFID, such tag design is lower cost and more accessible.

In addition to traffic, payload information provides additional ways to assess the health of the hive, for instance, by having a precise account of the pollen intake (Frias et al., 2016). The study of Babic et al. (2016) used low-resolution images and traditional image processing techniques for detecting pollen. Several convolutional neural network architectures were evaluated by Rodriguez et al. (2018a) on a higher resolution dataset of images of individual honey bees, showing the possibility of accurate automatic pollen detection. This approach was adapted to FPGA by Sledevič (2018) for low-cost real-time implementation. More recently, Marstaller et al. (2019) proposed a CNN architecture that can detect pollen balls on extracted images of individual bees.

2.2. Behavior Quantification From Video

More generally, behavior quantification is an extended field of research that has led to relevant developments for the monitoring of diverse animals from video.

Estimation and tracking of pose (position of the body parts) is a typical first step to classify and characterize actions and behavior. Even though most of the published study has focused on human pose estimation (Cao et al., 2017; Güler et al., 2018), several contributions have targeted animal behavior more specifically (Mathis et al., 2018; Nath et al., 2019).

Reliable long-term tracking is fundamental for the study of behavior. For tracking, Cascaded Pose Regression (Dollár et al., 2010) was applied to track individuals such as Drosophila, mice, and fish (Kabra et al., 2012). This method relied on an initial estimate to be refined progressively using a sequence of regressors. This type of approach is better suited for tracking in setups where individuals are always visible and captured at a high frame rate, as it requires an initialization when a new individual appears in the field of view.

Pose estimation built on top of deep network architectures was demonstrated to detect body part keypoints reliably in various animals (Mathis et al., 2018; Nath et al., 2019; Pereira et al., 2019). These methods however have so far focused on the tracking of a single animal. A recent exception was the detection and pose estimation of the honey bee body inside of the hive using a modified U-net architecture (Bozek et al., 2018). This approach estimated orientation and used a recursive formulation to guide detection in subsequent frames. As most bees were visible for extended periods of time, visual identity models were trained to solve reidentification when tracks crossed each other or an individual became occluded for a short amount of time.

Detailed pose estimation for multiple bees at the entrance of the hive was first shown in Rodriguez et al. (2018b). This last approach will be evaluated in detail and integrated into a complete system in this study.

3. MATERIALS AND METHODS

In this study, a complete system for foraging characterization of honey bee hives using recordings from the video is proposed. The system consists of the following modules: (1) data collection: recordings of video using camera capture at the colony entrance, (2) detection and tracking of honey bee individuals, (3) activity classification to decide if the bee is leaving, entering, or walking, and recognize the presence of a pollen payload, (4) identification of marked bees, (5) activity analysis through actograms summarizing extended periods of time. The general architecture is shown in **Figure 1**. The developed software is open-source and made available to the community as detailed in the *Data Availability Statement* section.

3.1. Data Collection

The video capture system is designed to observe the ramp through which all foraging bees must pass to exit or enter the colony. **Figure 2** shows the system used in this study. We used a 4 Mpixels GESS IP camera connected to a video recorder configured to record continuous H264 video at 8Mbps. A transparent acrylic plastic cover located on top of the ramp forces the bees to remain in the focal plane of the camera. To avoid interfering with the bee biological cycles, only natural light is used. A white plastic diffuses the natural light received, and a black mask is put around the camera to reduce the direct reflections that could be visible on the plastic cover.

The majority of videos used in this study were acquired from June 25 to June 30, 2017 at the UPR Agricultural Experimental

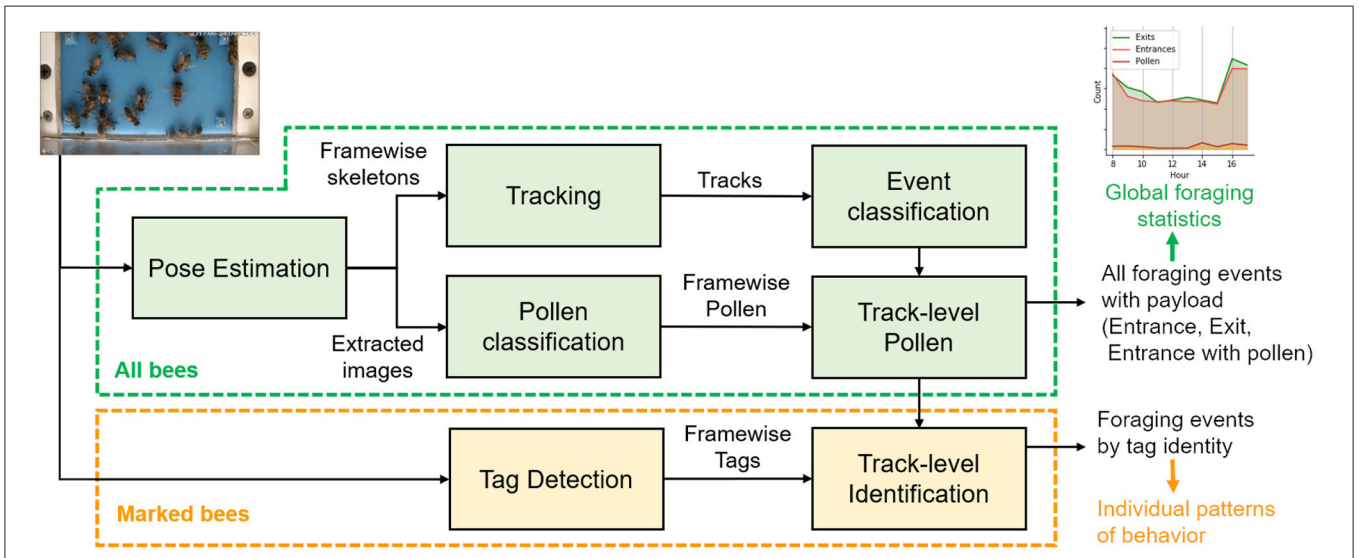


FIGURE 1 | Overview of the video analysis modules. In green, the base system can be applied uniformly to both unmarked and marked bees to detect all foraging events and produce aggregated statistics including global information of pollen intake. In orange, the identification module extends the base system to associate events to tagged bees to produce actograms for detailed analysis of behavior at the individual level.

Station of Gurabo, Puerto Rico, from 8 a.m.–6 p.m. to take into account different lighting setups. Each video is 1 h long.

3.2. Pose Estimation

Body pose is defined as a skeleton model combining a set of body parts with their connections. For honey bees, we considered five parts including Abdomen, Thorax, Head, AntL (left antenna), and AntR (right antenna), as shown in **Figure 3**. In the evaluation section, we will show an ablation study using simplified models with 3 parts (abdomen, thorax and head), 2 parts (abdomen and head), and 1 part (either thorax, or head). In terms of connections, the 5 parts model considers 5 connections: Head-Abdomen, Thorax-Abdomen, Head-Thorax, Head-AntL, Head-AntR. The simplified models consider the subset of connections that connect the considered parts.

We use a tracking-by-detection approach, where the individuals are first detected in each frame independently and then tracked through time to produce a pose trajectory, which provides for each individual bee the evolution over time of the position of their body part.

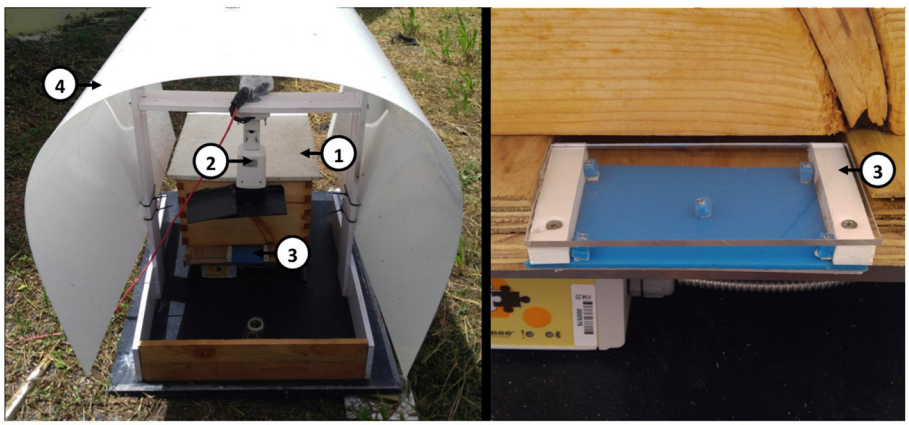


FIGURE 2 | Video capture system used in the field: (left) overview of the system installed at the entrance of the colony and (right) detail on the entrance. (1) Bee-hive, (2) camera, (3) entrance ramp, and (4) protection against direct sunlight.

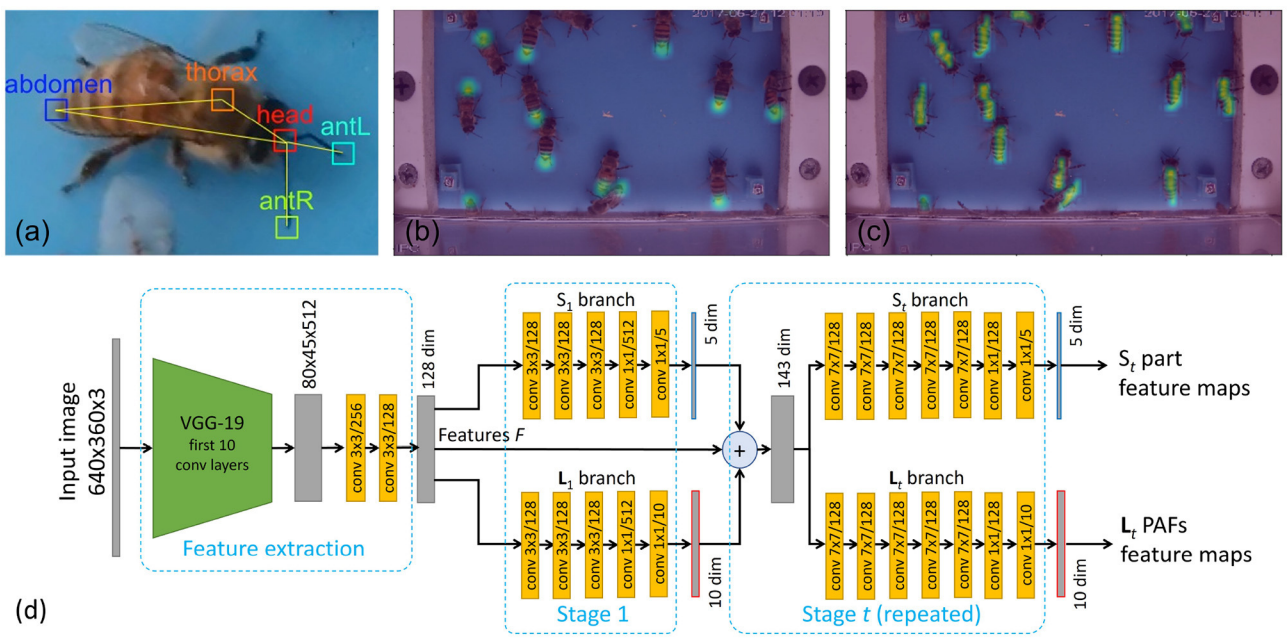


FIGURE 3 | Pose detection model. **(a)** 5 parts skeleton (Head, Thorax, Abdomen, Left and Right Antenna) and connection used. **(b)** Confidence map of the abdomen keypoints used to train the (S) branch. **(c)** The magnitude of the Part Affinity Field (PAF) of the thorax-abdomen connection used to train the (L) branch. **(d)** Convolutional Neural Network architecture used to predict the body part confidence maps and PAFs organized as a feature extraction backbone followed by the two branches (S) and (L) with iterative refinement stages.

3.2.1. Architecture of the Pose Detector

The pose estimation module in each frame follows the Part Affinity Fields (PAF) approach proposed by Cao et al. (2017), with modifications to take into account the specificity of the colony entrance videos. The convolutional neural network is composed of a feature extraction backbone (F) and two pose detection branches (S,L). Refer to **Figure 3d** for the architecture of the network used in this study.

Let us denote F as the feature map produced by the feature extraction backbone. Following Cao et al. (2017), we used the 10 first convolutional layers of VGG-19 (Simonyan and Zisserman, 2014), pre-trained on ImageNet. The next two layers (`conv4_3` and `conv4_4`) were modified to reduce the dimension from 512 down to 128 while keeping the same 1/8 resolution.

The keypoints branch (S) estimates a set of 2D confidence maps where each pixel belongs to a particular body part keypoint. Each confidence map is used to detect keypoints related to a single type of body part (e.g., head) for all individuals at once. The PAF branch (L) produces Part-Affinity Field vector maps that are used to connect body parts of different types. Each PAF is used to connect a single type of connection (e.g., head-thorax). The output of both branches is used as input of an inference algorithm outside of the network that estimates both the keypoints and their connections, to produce a set of individual body skeletons.

Let us now denote $S = (S_1, ..., S_j, ..., S_J)$ the set of J heatmaps, one per body part, and $\mathbf{L} = (\mathbf{L}_1, ..., \mathbf{L}_c, ..., \mathbf{L}_C)$ the C vector fields or PAFs, one per connection, considered for the configuration of the pose for each individual. The prediction of S and $\mathbf{L}$ is done with multiple stages of refinements producing multiple estimates S^t and $\mathbf{L}^t$. After a first stage $t = 1$ that takes as input the features F, subsequent stages $t > 1$ refine previous estimates S^{t-1} and $\mathbf{L}^{t-1}$ iteratively.

3.2.2. Training

Based on the annotated keypoints, the reference confidence map S_j^* is generated as the combination of a set of gaussian kernels centered on the x, y position of each annotated part of type j. Each channel of the (S) branch will be used for a different part. The reference PAF vector fields $\mathbf{L}_j^*$ are generated by the interpolation between two of the keypoints that belong to a joint. Channels of the (L) branch are organized in pairs of consecutive channels associated with a single connection. The reference skeleton model, confidence maps, and PAF fields are illustrated in **Figures 3a–c**.

The loss functions at stage t for confidence map j and PAF c are defined as the weighted mean squared errors $\sum_p W(p) \cdot ||S_j^t(p) - S_j^*(p)||_2^2$ and $\sum_p W(p) \cdot ||\mathbf{L}_c^t(p) - \mathbf{L}_c^*(p)||_2^2$, where S_j^* and $\mathbf{L}_c^*$ correspond to synthetic confidence maps and PAFs generated from the reference keypoints. $W(p)$ is a binary mask defined as 0 for all pixels that do not belong to the bounding box surrounding the keypoints of any annotated bee. The overall loss function is defined as the sum of the losses for all stages during training. Only the output of the last stage is used for inference.

To improve the generalization of the network, we performed data augmentation in the form of random geometrical transformations combining rotation, translation, reflection, and scaling applied to both the input image and associated reference keypoints. In order to maximize the usage of the training dataset, all generated images were centered on an individual, although the size was designed to also include significant contexts, such as other individuals and backgrounds.

3.2.3. Inference Stage

Once the set of confidence maps and PAFs are computed, the greedy inference is performed to (i) detect the body-part keypoints and (ii) group keypoints into skeletons illustrated in **Figure 4**.

Keypoints are detected by extracting the local maxima from the (S) confidence maps and applying non-maximum suppression with a fixed radius to remove duplicates. The detection threshold is fixed and only parts with a score higher than 0.5 are accepted as a hypothesis.

Grouping uses the PAF from the (L) branch to select the most likely connections by calculating the association score explained later in Equation (6). All the association candidates (j_1, j_2) with valid part types are sorted in descending score $E_{j_1 j_2}$ to perform

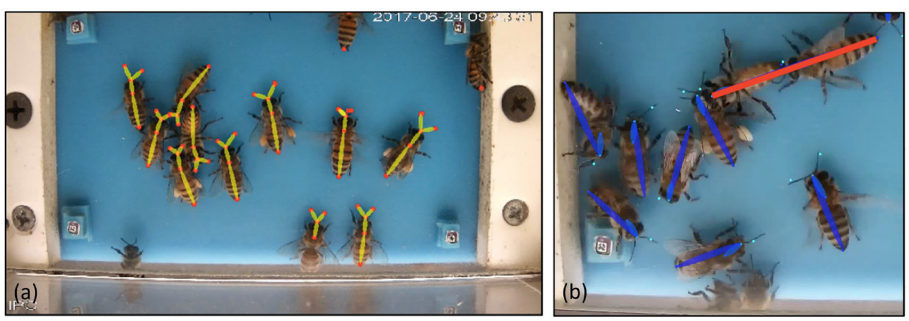


FIGURE 4 | Examples of bee skeletons reconstructed in the inference stage. **(a)** Because of the bottom-up greedy association, a partial skeleton can be generated even when some keypoints are not visible or detected. The use of our modified PAF score also enables the correct associations in crowded areas by constraining both orientation and length of the connections. **(b)** Example of incorrect association (in red) between head and abdomen obtained if factor $\pi_{j_1 j_2}$ is ignored when two bodies are aligned.

greedy bottom-up association. If a new association conflict with existing connections with either j_1 or j_2, it is discarded, else it is added to the connections. The final skeletons are obtained by extracting the connected components of all selected connections.

Given that the honey bees at the entrance may have completely arbitrary orientations, it is common for two or more individuals to be aligned. This is unusual in typical videos with humans, where people do not commonly have connections of the same type (e.g., abdomen-head) aligned with each other. The original approach measured the association between two parts by computing the line integral over the corresponding PAFs or, in other words, by measuring the alignment of the detected body parts. We used the distribution of the distance between points to constrain the connections based on the scale of the honey bees' bodies. The new PAF score associated with the body parts j_1 and j_2 is expressed as:

$$E_{j_1,j_2} = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{\mathbf{d}_{j_2} - \mathbf{d}_{j_1}}{\left\| \mathbf{d}_{j_2} - \mathbf{d}_{j_1} \right\|} \pi_{j_1 j_2} du \tag{1}$$

where the point $\mathbf{p}(u) = (1-u)\mathbf{d}_{j_1} + u\mathbf{d}_{j_2}$ moves along the linear segment between the two body parts for $u \in [0, 1]$. The probability $\pi_{j_1 j_2}$ is defined as the empirical probability that j_1 connects with j_2 conditioned on their distance $\left\| \mathbf{d}_{j_2} - \mathbf{d}_{j_1} \right\|$. It follows a gaussian distribution specific to each type of connection, fitted on the training dataset.

The π factor is important in scenarios where PAFs can be aligned since all individuals share the same PAF channels in the network. This particular assumption works well for honey bees, as their body is usually quite rigid, so the variance of the connection length is small. **Figure 4b** shows an example of alignment that leads to the incorrect association if this factor is omitted.

3.3. Tracking

Once the detections on each frame have been obtained, they are matched temporally on a frame-by-frame basis using the Hungarian algorithm (Kuhn, 1955). The cost used for the Hungarian assignment algorithm is determined by the distance between detected keypoints in a frame and the predicted positions from the past. We take into account not only point to point distance for the thorax but also antennae and head. Formally, the distance between two bees in consecutive frames is measured in the following way.

$$D(b_{k,i}, b_{k+1,i'}) = \frac{1}{J} \sum_{j=1}^{J} \delta(\mathbf{d}_{k,i,j}, \mathbf{d}_{k+1,i',j}) \tag{2}$$

where $b_{k,i}$ refers to the set of keypoints for detected bee i in frame k and $\mathbf{d}_{k,i,j}$ indicates the keypoint of part j for that bee, and J is the number of body parts considered. Finally, the cost $\delta(\mathbf{d}, \mathbf{d}')$ between two keypoints:

$$\delta(\mathbf{d}, \mathbf{d}') = \begin{cases} \|\mathbf{d} - \mathbf{d}'\| & \text{if both } \mathbf{d} \text{ and } \mathbf{d}' \text{ were detected} \\ \delta_0 & \text{else} \end{cases} \tag{3}$$

where δ_0 is a fixed misdetection penalty ($\delta_0 = 200$ pixels in our experiments) that is close to the typical distance between two bees located along with each other.

Once the assignment is done by the Hungarian method, an unassigned track will be considered a disappeared track and an unassigned detection in frame $i + 1$ will start a new track.

From that point on, detected bees can be associated with a trajectory index trackId instead of independent indices in each frame. We will denote by $T(i) = (T_{k,i}, T_{k+1,i}, ..T_{k',i})$ the trajectory with trackId i, starting at frame k, and ending at frame k'. Each element $T_{k,i}$ is associated with a detection $\mathbf{d}_{k,i}$, as well as optional frame levels information such as the presence of pollen or tag.

3.4. Foraging Events Detection and Labeling

3.4.1. Entrance/Exit Detection

We perform the detection of entrance and exit events using the starting and ending points of the tracks using the classification policy illustrated in **Figure 5**.

First, tracks shorter than 5 frames are discarded. The leaving and entering behavior is defined by crossing the red line (as shown in **Figure 5**). The direction of the track would determine if it is leaving or entering. Tracks that both start and end in the middle are considered walking bees. The other two classes of tracks are those that start and end in the bottom (outside) or top (inside). These tracks are among those bees that dwell in the border and enter and exit the field of view of the camera without actually crossing or entering the center of the ramp. Although these tracks could be used in the future to assess the level of crowding of the ramp, they are ignored in this study to focus on the entrance and exits.

3.4.2. Pollen Classification at Frame and Track Levels

Once the pose estimation is performed, the location of the keypoints of the head and the tip of the abdomen are used to precisely crop the image of each individual bee with optional orientation compensation, as shown in **Figure 6**. The image obtained is then passed to a shallow network with two layers that have previously shown high accuracy in pollen detection classification (Rodriguez et al., 2018a) to predict the decisions for each detection in each frame.

Track-level pollen classification is performed by aggregating the framewise pollen classification using a majority vote.

3.5. Bee Identification Using Tags

Once all bees have been detected and tracked, the individual identity information can be added to the tracks for those bees that have been marked with a tag.

The tag detection is performed with the help of the open source project April Tags (Wang and Olson, 2016). We have generated the tag25h5 family, which consists of 5 x 5 binary codes with a minimum Hamming distance of 5 between codes, and that code for 3009 unique tags. The tags are printed on waterproof paper using a 1,200 dpi laser printer, cut automatically using a laser cutter, and placed on the torso of the honey bees. The code used to generate the tag sheets for printing and laser cutting

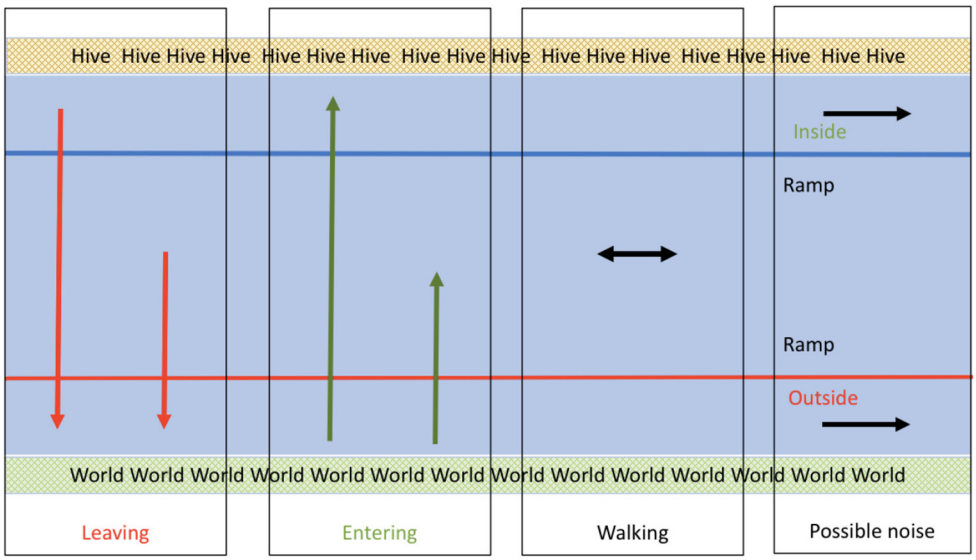


FIGURE 5 | Event classification policy based on trajectory endpoints. The entrance to the hive is at the top and the outside world is at the bottom. Red arrows represent exit events (leaving): these trajectories start inside the ramp and end below the threshold represented by the red line. Green lines represent entrances: these trajectories start at the bottom and end inside the ramp. Black lines represent ignored trajectories that are the result of tracking interruptions or of bees that dwell in a single area but never enter nor exit.

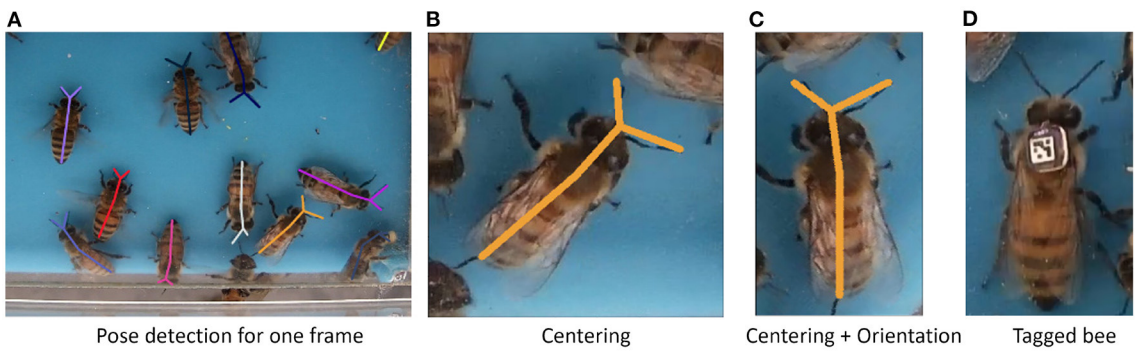


FIGURE 6 | Detail of individual bees. **(A)** Typical frame with a detected pose. **(B)** Cropped image with thorax centering. **(C)** Cropped image with thorax centering and orientation compensation using the head and abdomen axis. **(D)** Example of bee tagged with a 5 × 5 apriltag barcode of the tag25h5 family.

is provided in the *generatetagsheet* package shared in the *Data Availability* section.

In contrast to existing systems for individual monitoring (Crall et al., 2015) or (Boenisch et al., 2018) that are based primarily on tag detection, we consider tags as an augmentation of the trajectories obtained from whole body detection, thus tracking marked and unmarked bees with the same process. Given that tags are much smaller than the honey bee themselves, they can be more easily occluded, and their high-resolution barcode is more easily impacted by motion blur. For these reasons, we chose to perform whole bee detection as the main input for tracking, even for marked bees. Our system, therefore, relies on the assumption that the framerate is fast enough to track without ambiguity based on geometric proximity alone (we used 20 frames per second in our experiments). In each frame k, each trajectory element $T(k, i)$ is then potentially associated with a tag ID by finding the detected tag whose center is the closest to the thorax keypoint. In case the distance is larger than a threshold $d > 0$, the track is not assigned any tag for this frame. After association, the most frequent tag ID is selected for each track, thus augmenting each trajectory with individual identity when available.

4. RESULTS

The system has been evaluated with respect to two main aspects: (i) pose detection in individual frames, and (ii) foraging event detection and classification (entrance/exit, presence of pollen).

4.1. Performance of Detection and Pose Estimation

4.1.1. Dataset for Pose Estimation of Honey Bees

As part of the contributions of this study, a *honey bee pose dataset* is released. This dataset consists of 270 frames, chosen from a pool of 10 videos distributed across several days between June 22, 2016 and June 28, 2016, and different times of the day, using the recording setup described in Section 3.1. Each frame was fully labeled using 5 key-body points (Head, Thorax, Abdomen, AntL, and AntR). The dataset contains a total of 1,452 fully labeled bees. The annotations are stored in COCO format (Lin et al., 2014).

4.1.2. Evaluation Methodology

The evaluation of the algorithm was performed using mean Average Precision (mAP) as provided by the pose evaluation package https://github.com/leonid-pishchulin/poseval based on Insafutdinov et al. (2017). The mAP metric relies on the greedy assignment of the body pose predictions to the ground truth (GT) based on the highest PCKh. PCKh-0.5 is the standard metric used for human pose estimation where a keypoint is correctly detected if its residual distance to the ground truth is less than half of the head size (Andriluka et al., 2014). As an equivalent for honey bee pose estimation, we considered that a key-point is correctly detected if its residual is less than half of the distance between thorax and head. We defined a fixed threshold for all the individual detections of 0.5, thus making sure that only high confidence keypoint hypotheses were passed to the inference stage.

The dataset was split into training and validation datasets as follows: 70% of the data was used for training and the remaining 30% for validation. The frames used for validation were extracted from videos captured on a different day and at different times of the day than the training to ensure as much independence as possible between the datasets. The VGG19 feature extraction backbone was frozen with weights pre-trained on ImageNet. The training was performed on the S and L branches using the Adam optimizer with a learning rate of $1e^{-4}$.

Figure 7 shows an overview of the performance for 1, 2, 3, and 5 parts and 2, 3, 4, 5, and 6 stages models. We trained the model for 20,000 epochs, and show the mAP averaged between 19,000 and 20,000 epochs. These results are analyzed in the next subsections.

4.1.3. Effect of the Number of Parts

Figure 7 and **Table 1** show that in terms of detection mAP, higher scores are obtained with the 5 parts model than the models with fewer parts. We hypothesize that the higher number of parts may help the network interpolate poorly detected parts by using features learned for the detection of its connected parts. This is supported by the fact that the head part benefits more from the addition of the antennas, to which it is connected in the 5 parts model, compared to the thorax, which is not connected to the antennas in any model.

We also compared our approach with the popular YoloV3 object detector (Redmon and Farhadi, 2018), trained to detect a bounding box around the thorax. Such a detector could be used in a two-step approach where each individual is first detected approximately, before applying a more precise body parts model on each individual separately. This top-down 2-steps approach is necessary to apply the body parts models designed for one individual such as Mathis et al. (2018), Marstaller et al. (2019), and Pereira et al. (2019). The bounding boxes used for training were centered on the thorax ground-truth with a fixed size of 150 × 100 pixels. The detector was trained until early stopping with standard parameters of 9 anchor boxes, 13 x 13 grid size, and pre-trained weights from Darknet-53. The model used a standard threshold of 0.5 to discard overlapping anchor boxes. For testing, the center of the bounding box was used as the estimate of the thorax location for comparison with keypoint detection. For these comparisons, we used the implementation by Zihao Zhang from https://github.com/zzh8829/yolov3-tf2. In our experiments, Yolo only reaches 79.8 mAP, compared to 97.6 when we train our approach for the detection of only the thorax part. Based on these results, we did not consider approaches based on anchor boxes further for the proposed monitoring system, relying instead on the more precise heatmap based keypoint detection.

4.1.4. Effect of the Refinement Stages

The refinement stages proposed in Cao et al. (2017) are designed to increase the accuracy of pose estimation. A rationale is that due to the difficulty of the task, and the diversity of conditions, scale, background, part detection, and affinity fields may not be estimated properly and/or in a consistent way in the first stage. Recursively refining the part detection and affinity fields was shown to improve the performance in human pose estimation. We now explore the effect of the refinement stages for honey bee detection.

The results from **Figure 7** show no clear trend of the performance in terms of the number of stages. This is confirmed in **Table 2** where mAP performance for all keypoints is shown in the case of the 5 parts model. In our case, the videos have a single background and all the animals preserve their scale with respect to the camera, which could help in obtaining good performance without the need for extra refinement stages. The two best performing number of stages are 2 and 6, with a small decrease of performance for the intermediate number of stages. Since each additional stage adds 9 extra layers, we, therefore, decided to limit the model to 2 stages in further experiments, which allowed us to allocate more models in parallel inside the GPU and to increase the throughput of the pipeline.

4.2. Performance of Foraging Events Detection

Foraging events correspond to entrances and exits of honey bees in the ramp with the potential presence of pollen. We evaluated the different aspects of their detection as follows: tracking in Section 4.2.1, entrance/exit detection in Section 4.2.2, and pollen recognition in Section 4.2.3.

4.2.1. Tracking

For the evaluation of tracking, a video segment of 1,200 frames (60 s) was fully annotated manually. During the manual annotation, it was identified the intrinsic difficulty of creating

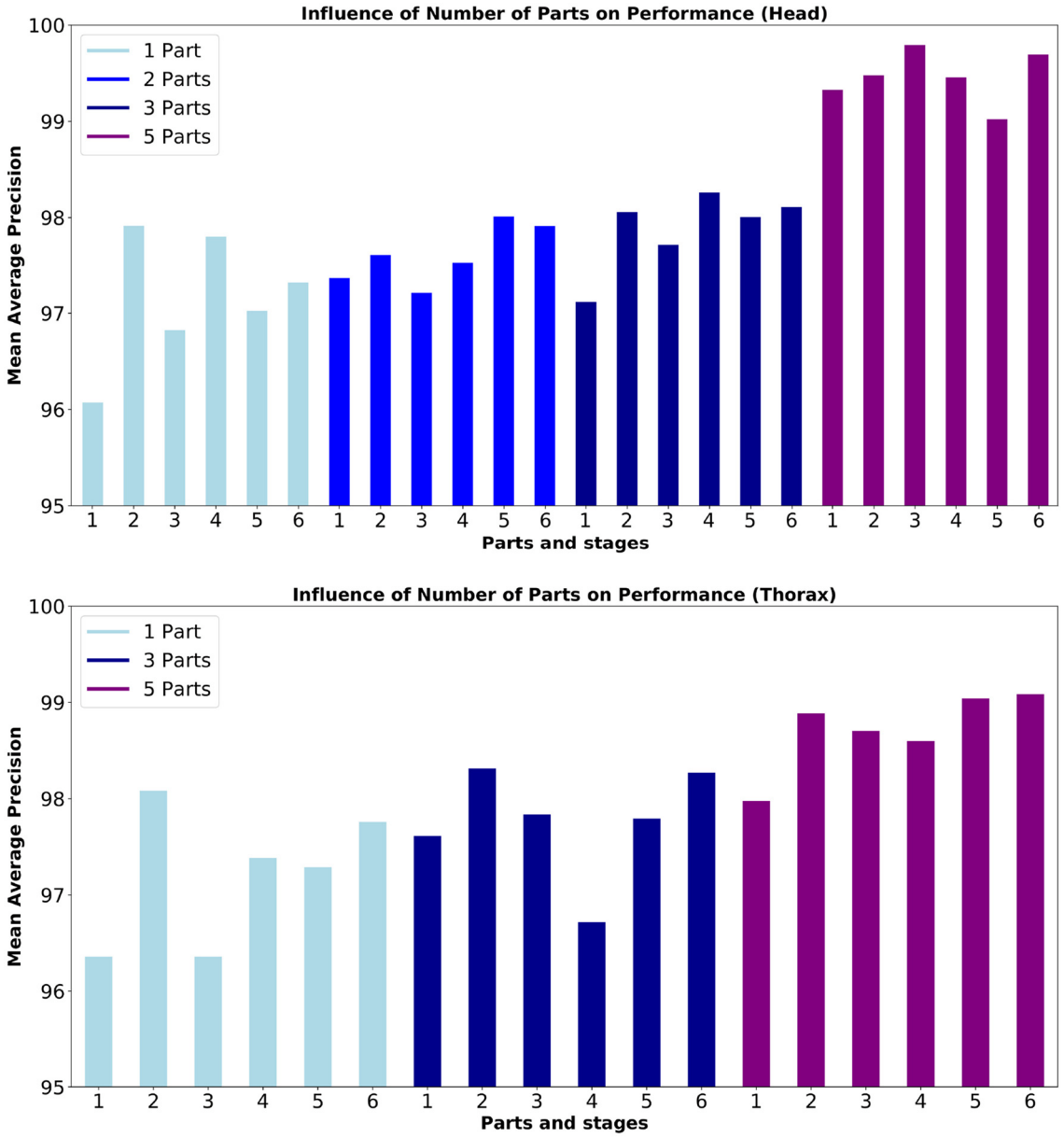


FIGURE 7 | Mean Average Precision (mAP) for the detection of (top) head and (bottom) thorax. The box plot shows performance mean 19,000 epochs of training. From left to right, results for Head show training using 1 part (Head), 2 parts (Head, Abdomen), 3 (Head, Abdomen, Thorax), and 5 parts (3 parts + 2 antennas) results for Thorax show training using 1 part (Thorax), 3 parts (Head, Abdomen, Thorax), and 5 parts (3 parts + 2 antennas). Description of the mAP can be found in Section 4.2.

the groundtruth close to the borders of the field of view, where individuals are only partially visible and it is not clear where to stop tracking and how to identity fragmented tracks. For example, a bee partially leaving the field of view may still be visible enough for a human annotator to keep track with a consistent identity but only by using uncertain visual cues. For this reason, all annotations were filtered to keep only detections and track fragments located in the interval $y \in [200, 1{,}000]$ pixels, which includes the inside and outside thresholds $y = 600$ and $y = 300$. A large portion of the excluded data corresponds to static bees remaining close to the entrance, or bees walking near the edge, which are not relevant for entrance/exit detection. The corresponding dataset contains 6,687 detections assigned to 79 tracks.

The tracking algorithm was configured with a maximum matching range of 200 pixels. The evaluation was performed using the package from https://github.com/cheind/py-motmetrics for multiple object tracker metrics. All ground-truth tracks were mostly tracked (tracked for more than 80% of the track length). At the detection level, it was identified 0 false

TABLE 1 | Summary of the best mAP when training on skeletons composed of 1, 2, 3, and 5 parts and YOLO-v3 on Thorax.

	Yolo(%)	1 part*(%)	2 parts(%)	3 parts(%)	5 parts(%)
Head	–	96.3	98.7	96.4	99.1
Tip abdomen	–	–	94.0	96.2	95.0
Thorax	79.8	97.6	–	95.0	98.7
Right antenna	–	–	–	–	94.4
Left antenna	–	–	–	–	90.4

**Two different models were used for 1 part: one model with the head only, another for thorax only.*

TABLE 2 | Body part detection Performance (mAP) per number of stages.

	1 stage AP(%)	2 stages AP(%)	3 stages AP(%)	4 stages AP(%)	5 stages AP(%)	6 stages AP(%)
Head	97.3	98.4	96.5	97.1	96.4	98.2
Tip abdomen	97.6	96.1	92.6	94.6	99.0	99.1
Thorax	96.2	97.0	95.6	97.6	99.2	99.7
Right antenna	90.3	90.7	87.6	84.4	86.1	94.4
Left antenna	86.5	87.7	88.6	85.8	88.1	90.4

positives and 20 misdetections out of 6,687, which generated 10 track fragmentations. A total of 12 identity switches were found, mostly due to this fragmentation.

In the future, we plan to use visual identity models (Romero-Ferrero et al., 2018) integrated into more complex predictive tracking algorithms to reduce such fragmentation and help with the re-identification of individuals that exit the field of view for a short time.

4.2.2. Entrances and Exits

We used data captured on June 27, 2017 (10 h of video from 8 a.m.–6 p.m.) to evaluate entrance and exit detection performance. Manual annotation was performed on tagged bees to label each tagged bee track as entering or leaving, ignoring walking bees. This approach was used to facilitate human annotation, as the tag could be used by the annotators to make sure they inspected the behavior of each individual for the complete interval in which they were in the field of view without relying on the tracking algorithm itself.

The automatically detected body pose tracks (which does not rely on the presence of a tag) were augmented with the information from the manual annotations. The automatic entrance/exit event classification of these tracks was compared to the corresponding manual annotation. Automatic detection could also produce *Noise*, which would be a track that does not cross the entrance virtual line.

The confusion matrix in **Table 3** shows that overall entrances perform better than exits. There is a low number of *Noise* for entrances too, which seems to be related to the fact that bees returning to the colony do not typically stay on the ramp, and enter the colony immediately. For exits, there is a much higher number of *Noise* trajectories. Unlike bees that are entering, exiting bees can spend more time on the ramp, which makes them vulnerable to track interruption due to misdetection on any of these frames. These interruptions would warrant an improvement of the tracking algorithm to reduce interruptions and with additional re-identification of lost tracks in the future.

TABLE 3 | Confusion matrix of entrance and exit detection. Rows represent the manual annotation of each corresponding track (groundtruth).

GT\DT	Entrance	Exit	Noise	Total
Entrance	99	3	4	106
Exits	3	83	14	102

Each column represents the automatic classification (detection).

4.2.3. Pollen Recognition

We carried out the evaluation of pollen detection at two levels: first, at the image level using a manually annotated image dataset, and second, at the event level using manually labeled entrance events. The image level classifier in both cases follows previous study (Rodriguez et al., 2018a) discussed in the *Method* section.

4.2.3.1. Evaluation of Pollen Recognition at the Image Level

The image dataset consists of 1,550 annotations of pollen (775) and non-pollen (775) bees from 3 videos, collected at 12 p.m., 1 p.m. and 2 p.m. on the 22, 23, and 24 of June 2017, respectively. Each pollen bearing bee entering was annotated only once, as well as another non-pollen bearing bee in the same frame that served to balance the dataset with a negative sample taken in similar conditions. For training and validation, an 80–20 random split was used. This dataset provides about double the amount of annotation than a previously released dataset (Rodriguez et al., 2018a) and will be shared as well.

We used three different methodologies for extraction of the cropped images of each individual which was fed as input of the pollen classification network. The first approach, *Manual centering*, consists of the extraction of the images based on the thorax position and creating a box of 450 x 375 pixels around it. The second one, *Manual centering and orientation*, uses the manually annotated position of the head and abdomen to rotate and center the image so that all extracted 300 x 200 pixels images are aligned vertically and centered on the midpoint between head and abdomen. The last approach, *Automatic centering and orientation*, was the same as the previous one but used the detected head and abdomen parts instead of the manually annotated ones. The groundtruth of pollen labeling was in this case obtained by matching the manual thorax annotation to the closest automatic thorax detection.

In the results shown in **Table 4**, we observe a 3% increased performance when using orientation compensated images. Coupled with previous results, which showed that the 5-parts model performed better than simpler models of detection, this supports the use of the proposed detection based on a pose model. which both leverages the annotations of different body parts to improve detection and provides direct information to compensate orientation and facilitate the down-stream analysis such as pollen detection. Pose compensation using

automatic pose detection obtained slightly better performance than compensation using manually annotated data. Visual inspection revealed that human annotation was more imprecise when annotating such a large amount of frames, while automatic centering produced slightly more consistent alignment.

4.2.3.2. Evaluation of Pollen Recognition at the Event Level

We used the same three models for image level classification as in the previous subsection (Manual centering, Manual centering and orientation, and Automatic centering and orientation). Track level classification was performed as explained in the Methods section.

The results are summarized in **Table 5**. The best performance is obtained by using the model with automatic centering and orientation compensation, followed by manual compensation with a 2% difference. The automatic compensation makes sure all the bodies are aligned, thus, facilitating the pollen classification task. Centering only experienced a higher drop in performance, since the classifier has the harder task to learn invariance to orientation during training, which is not needed if orientation is compensated explicitly. This supports the proposed bottom-up approach where orientation is estimated as part of the detection step.

The 81.2% precision corresponds to a mere 21 false positives, which impact severely the metrics due to the high imbalance in the dataset (100 positives vs. 1,150 negatives). In terms of recall, a visual inspection of the errors showed that imperfect tracking accounts for most of the 22 false negatives presented. Improving tracking robustness is, therefore, an important aspect for future study to be able to improve the accuracy of the estimates.

TABLE 4 | Performance of pollen detection at the image level.

Cropping method	Precision(%)	Recall(%)	F_1
Manual centering only	94.58	94.56	0.946
Manual centering + orientation	96.30	96.25	0.963
Automatic centering + orientation	97.26	97.28	0.973

TABLE 5 | Performance of the detection of the event "entrance with pollen."

	Precision(%)	Recall(%)	F_1
Manual centering	59.7	52.0	0.555
Manual centering + orientation	80.1	75.0	0.776
Automatic centering + orientation	81.2	78.1	0.798

4.3. Application to Long-Term Monitoring

As a proof-of-concept of the scalability of the approach, the proposed system was adapted for high-performance computing platforms and applied on multiple days of videos.

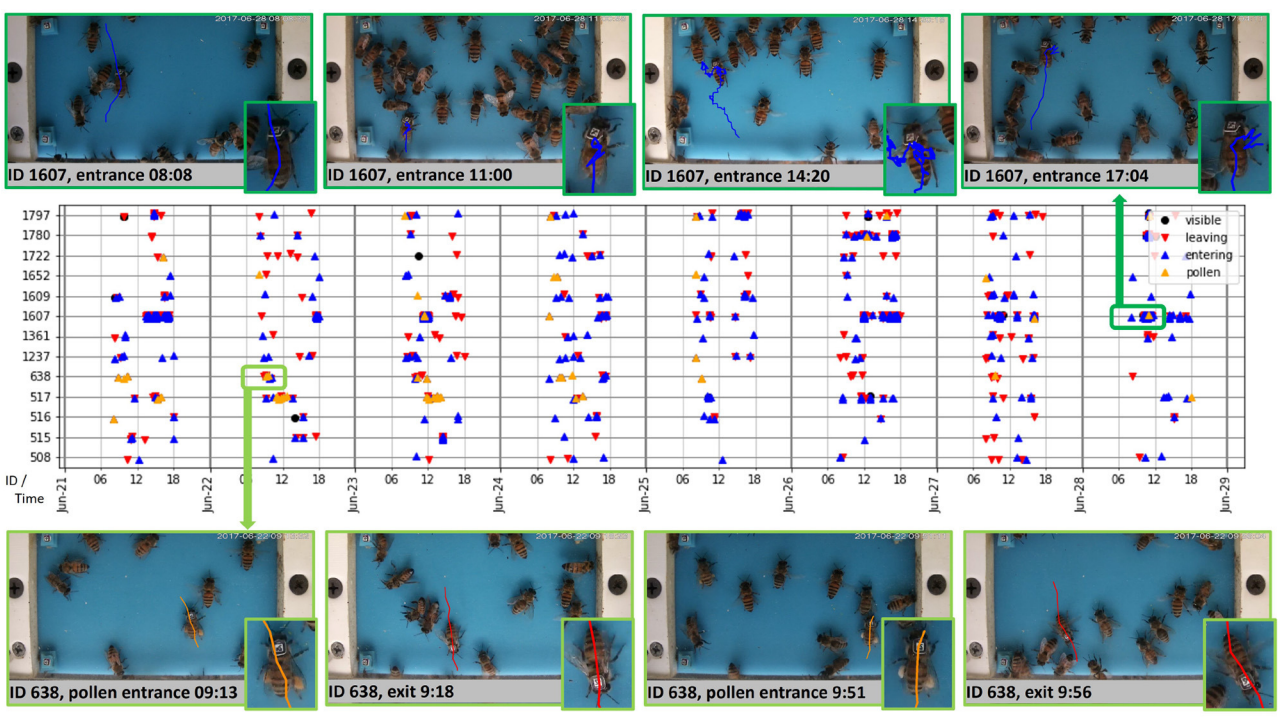


FIGURE 8 | Timeline of foraging events for several tagged bees over the span of 8 days (Middle), with the example of trajectories for bee #1607 (Top) and bee #638 (Bottom). Video analysis was performed from 8:00 to 17:59 each day from June 21 to June 28, 2017. The timeline shows the individual patterns of behavior of each bee over multiple days. The images show a sampling of event types (entrance, entrance with pollen, exit), luminosity conditions (after sunrise, daylight, before sunset), and ramp crowding (8–30 bees). The trajectories shown in these plots are based on the center of the body which may be slightly shifted from the tag position.

4.3.1. Computational Complexity Evaluation

Profiling of the code was done on a node at UPR HPCf with the following specifications: Intel Xeon E5-2630 v4 CPU at 2.20GHz and 128MB RAM equipped with 1 NVIDIA Tesla P100 GPU card with 12MB memory. The analysis was performed on a typical video of the dataset with 72,000 frames (resolution 2,560 x 1,440, 20 fps) resized to 1/4 its original size (640 x 360). The offline processing could be performed at a speed equivalent to ˜13.8 fps, with 88% of the time spent in the detection and 10% in pollen classification, tracking, and other processing adding negligible time.

Based on these results, we expect further optimization of the network architecture (such as replacing the VGG backbone with a more recent lightweight backbone) and its numerical implementation should enable real-time processing on GPU accelerated edge devices to enable deployment directly in the field.

4.3.2. Multi-Day Dataset

The dataset is composed of 8 days of daylight videos acquired from 8 a.m.–5 p.m., representing more than 72 h of video. Three types of events were detected: exit, entrance without pollen, and entrance with pollen. **Figure 8** shows several examples of such events for a few tagged bees, in relation to the total timeline that shows the individual behavior.

In this dataset, we used April Tags (Wang and Olson, 2016) that include error correcting codes. The tags printed from the tag25h5 family (refer to **Figure 6D**) appeared with a size of around 25 × 25 pixels in the video frames. In these conditions, the identification was relatively unambiguous, as 86% of the tracks had all their associated tags with the same ID, and only 5% of the remaining tracks (0.7% of all tracks) had the majority ID account for less than 90% of the associated tags. In more challenging conditions, the presence of conflicting IDs may be used to monitor identification errors.

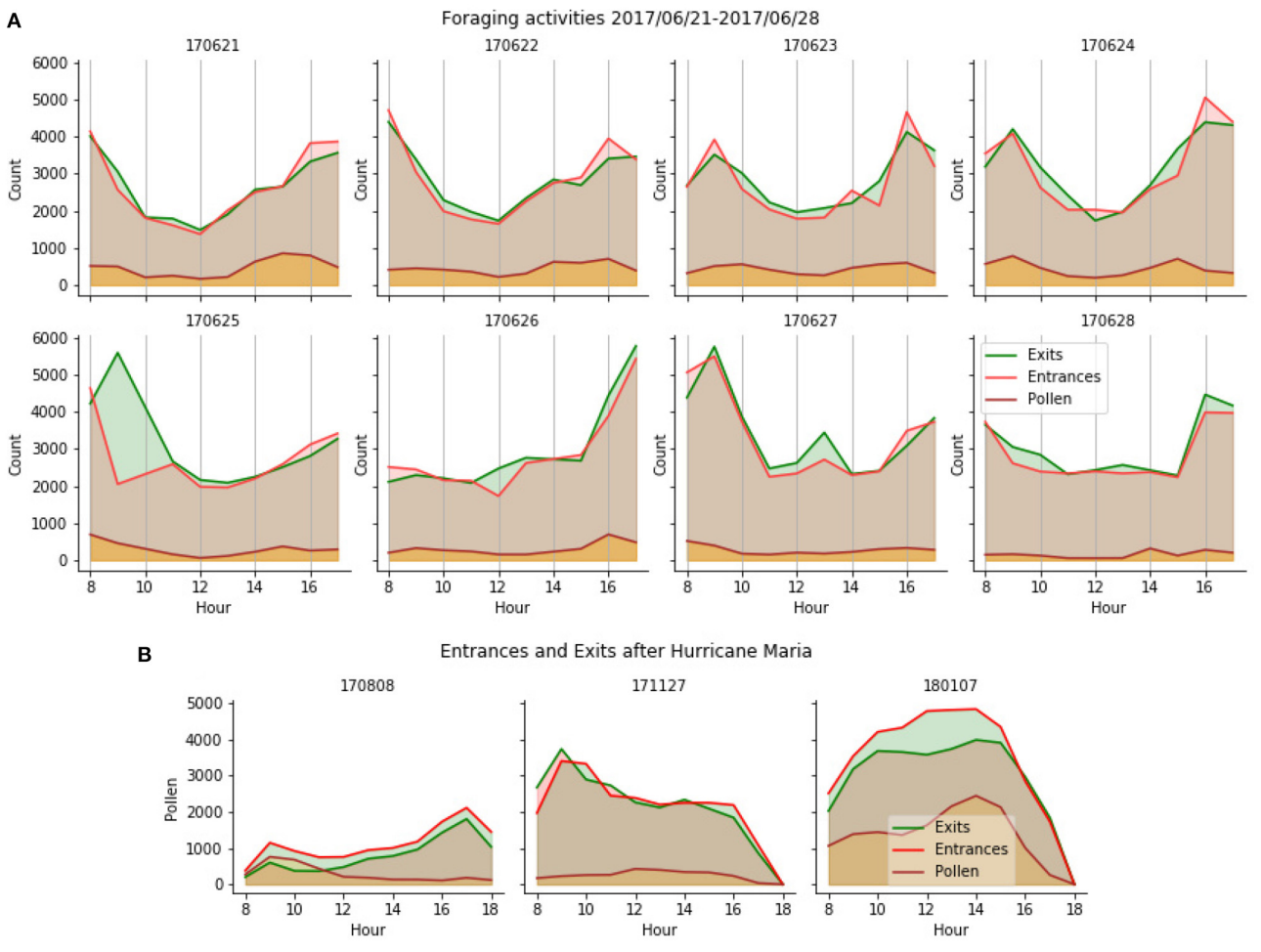


FIGURE 9 | (A) Hourly counts of entrance and exits for 8 consecutive days. Entrances with pollen, overlayed as a separate statistic, appear to not be directly proportional to the total entrances. The peak at 9 a.m. on 2017/06/25 is due to a temporary misalignment of the camera that occurred during maintenance of the colony. **(B)** Hourly counts before (2017/08/17) and after (2017/11/27, 2018/01/07) hurricane Maria show very diverse foraging patterns (refer to text).

Because the system tracked all bees, with or without a tag, the global foraging activity of the colony was also measured. The 3 types of events were counted and aggregated in 1-h intervals to produce the actograms shown in **Figure 9**. The top panel shows results on the same 8 days as in **Figure 8**. The bottom panel shows results on three additional days, at different times of the year including pre- and post-hurricane Maria (August 2018 and November 2018, respectively). The automatic detection of foraging patterns aligned with the lower activity at the end of the summer and the scarce sources of food that happened after the flora in the island was impacted by the hurricane. The much higher activity in pollen foraging observed a few weeks later in January 2018 can be explained by the rising food sources availability and the ability of the colony to reproduce, which requires proteins provided by pollen.

5. DISCUSSION

In this study, we presented a new system for the automatic surveillance of honey bees at the hive entrance using machine learning and computer vision and applied them to implement an end-to-end pipeline that quantifies their foraging behavior.

First, we presented an adaptation of the Part Affinity Fields approach for the detection and tracking of honey bees. Results showed that this tracking-by-detection approach produces high-quality results in presence of multiple individuals and is a promising approach to obtain precise estimates of pose for behavioral studies. We presented an ablation study of this architecture, showing the effect of the number of stages and the number of parts in the quality of the detection.

The precise detection of body parts allowed us to create tracks for all bees, both marked and unmarked while providing the identity of the marked bees when they appeared. This constitutes a multi-resolution view of the activity of the colony as specific behaviors patterns could be assigned to individual marked bees over long periods of time while capturing the global statistics of the behavior of unmarked bees.

We also compared several methods for pollen bearing recognition in honey bees. We concluded that using automatic alignment from the pose estimation and a convolutional neural network for image classification improved the accuracy, supporting the use of a detailed pose model.

These methods were combined to detect and characterize foraging behavior in honey bees. Application to videos capturing multiple days, and covering different times of the day and different parts of the year showed the applicability of the approach for large-scale offline analysis.

This system constitutes the first complete system that accounts for the foraging behavior that includes pollen foraging at global and individual levels. It provides a platform that can be built upon in the future to account for other behaviors such as fanning, paralysis (Bailey, 1965), or the presence of parasites such as varroa-mite (Bjerge et al., 2019). Based on this prototype, we are currently working on the computational and architectural optimization necessary to obtain real-time operation in the field. To scale the approach further, we are also considering extending re-identification beyond tag recognition. This would benefit from the availability of detailed pose information extracted by the system and simplify the logistics of marking a large number of bees to access individual behaviors.

DATA AVAILABILITY STATEMENT

The code and annotated datasets used for this study can be found at https://github.com/piperod/beepose, https://github.com/jachansantiago/plotbee, https://github.com/rmegret/apriltag, and https://github.com/rmegret/generatetagsheet.

AUTHOR CONTRIBUTIONS

IR contributed to the design, implementation, and evaluation of the methods and system. JC contributed to the implementation, scaling, and evaluation of the system. MA contributed to the evaluation of pose estimation. KB contributed to the design of pose estimation. JA-R contributed to the design and coordination of data collection and annotation. TG contributed to the design of data collection. RM contributed to the design and implementation of the system, and coordinated the study. All authors contributed to the article and approved the submitted version.

FUNDING

This material is based upon work supported by the National Science Foundation (NSF) under awards numbers 1707355 and 1633184. JC and MA acknowledge support from the Puerto Rico Louis Stokes Alliance For Minority Participation (PR-LSAMP), a program from the NSF under awards numbers HRD-1906130 and 1400868. TG acknowledges a National Science Foundation (NSF) OISE award (#1545803) and NSF-DEB award (#1826729) that provided for research release after hurricane Maria and NSF-HRD award (#1736019) provided funds for the purchase of bees. IR acknowledges support from HHMI Janelia Research Campus as an invited researcher. This study used the High-Performance Computing facility (HPCf) of the University of Puerto Rico, supported by the National Institute of General Medical Sciences, National Institutes of Health (NIH) award number P20GM103475 and NSF grant nos. EPS-1002410 and EPS-1010094. This study used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant no. ACI-1548562. Specifically, it used the Bridges and Bridges-2 systems (Towns et al., 2014; Nystrom et al., 2015), which are supported by NSF awards numbers ACI-1445606 and OCI-1928147, at the Pittsburgh Supercomputing Center (PSC).

REFERENCES

Anderson, K. E., Sheehan, T. H., Eckholm, B. J., Mott, B. M., and DeGrandi-Hoffman, G. (2011). An emerging paradigm of colony health: microbial balance of the honey bee and hive (apis mellifera). *Insectes Sociaux* 58, 431. doi: 10.1007/s00040-011-0194-6

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). "2D human pose estimation: New benchmark and state of the art analysis," in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 3686–3693.

Babic, Z., Pilipovic, R., Risojevic, V., and Mirjanic, G. (2016). Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* 7, 51–57. doi: 10.5194/isprsannals-III-7-51-2016

Bailey, L. (1965). Paralysis of the honey bee, apis mellifera linnaeus. *J. Invertebrate Pathol.* 7, 132–140. doi: 10.1016/0022-2011(65)90024-8

Bjerge, K., Frigaard, C. E., Mikkelsen, P. H., Nielsen, T. H., Misbih, M., and Kryger, P. (2019). A computer vision system to monitor the infestation level of varroa destructor in a honeybee colony. *Comput. Electron. Agric.* 164:104898. doi: 10.1016/j.compag.2019.104898

Boenisch, F., Rosemann, B., Wild, B., Dormagen, D., Wario, F., and Landgraf, T. (2018). Tracking all members of a honey bee colony over their lifetime using learned models of correspondence. *Front. Rob. AI* 5:35. doi: 10.3389/frobt.2018.00035

Bozek, K., Hebert, L., Mikheyev, A. S., and Stephens, G. J. (2018). "Towards dense object tracking in a 2d honeybee hive," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4185–4193.

Branson, K., Robie, A. A., Bender, J., Perona, P., and Dickinson, M. H. (2009). High-throughput ethomics in large groups of Drosophila. *Nature Methods* 6, 451–457. doi: 10.1038/nmeth.1328

Campbell, J., Mummert, L., and Sukthankar, R. (2008). "Video monitoring of honey bee colonies at the hive entrance," in *Workshop Visual Observation and Analysis of Vertebrate and Insect Behavior (VAIB) at International Conference on Pattern Recognition (ICPR)* (Tampa, FL), 1–4.

Cao, Z., Simon, T., Wei, S., and Sheikh, Y. (2017). "Realtime multi-person 2d pose estimation using part affinity fields," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 1302–1310.

Chen, C., Yang, E.-C., Jiang, J.-A., and Lin, T.-T. (2012). An imaging system for monitoring the in-and-out activity of honey bees. *Comput. Electron. Agric.* 89, 100–109. doi: 10.1016/j.compag.2012.08.006

Chiron, G., Gomez-Krämer, P., and Ménard, M. (2013). Detecting and tracking honeybees in 3d at the beehive entrance using stereo vision. *EURASIP J. Image Video Process.* 2013, 59. doi: 10.1186/1687-5281-2013-59

Crall, J. D., Gravish, N., Mountcastle, A. M., and Combes, S. A. (2015). BEEtag: a low-cost, image-based tracking system for the study of animal behavior and locomotion. *PLoS ONE* 10:e0136487. doi: 10.1371/journal.pone.0136487

de Souza, P., Marendy, P., Barbosa, K., Budi, S., Hirsch, P., Nikolic, N., et al. (2018). Low-cost electronic tagging system for bee monitoring. *Sensors (Basel)* 18, 2124. doi: 10.3390/s18072124

Dollár, P., Welinder, P., and Perona, P. (2010). "Cascaded pose regression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (San Francisco, CA: IEEE), 1078–1085.

Frias, B. E. D., Barbosa, C. D., and Lourenço, A. P. (2016). Pollen nutrition in honey bees (apis mellifera): impact on adult health. *Apidologie* 47, 15–25. doi: 10.1007/s13592-015-0373-y

Giannoni-Guzmán, M. A., Rivera-Rodriguez, E. J., Aleman-Rios, J., Melendez Moreno, A. M., Pérez Ramos, M., Pérez-Claudio, E., et al. (2021). The role of colony temperature in the entrainment of circadian rhythms of honey bee foragers. *Ann. Entomol. Soc. Am.* 114, 596–605. doi: 10.1093/aesa/saab021

Güler, R. A., Neverova, N., and Kokkinos, I. (2018). "Densepose: dense human pose estimation in the wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7297–7306.

Huang, Z. Y., and Giray, T. (2012). Factors affecting pollinators and pollination. *Psyche* 2012:e302409. doi: 10.1155/2012/302409

Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., et al. (2017). "ArtTrack: articulated multi-person tracking in the wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 1293–1301.

Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S., and Branson, K. (2012). JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* 10, 64–67. doi: 10.1038/nmeth.2281

Kaspar, R. E., Cook, C. N., and Breed, M. D. (2018). Experienced individuals influence the thermoregulatory fanning behaviour in honey bee colonies. *Animal Behav.* 142, 69–76. doi: 10.1016/j.anbehav.2018.06.004

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Res. Logistics Q.* 2, 83–97. doi: 10.1002/nav.3800020109

Kulyukin, V., and Mukherjee, S. (2019). On video analysis of omnidirectional bee traffic: counting bee motions with motion detection and image classification. *Appl. Sci.* 9, 3743. doi: 10.3390/app9183743

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft COCO: common objects in context," in *European Conference in Computer Vision* (Zurich), 740–755.

Magnier, B., Ekszterowicz, G., Lauren, J., Rival, M., and Pfister, F. (2018). "Bee hive traffic monitoring by tracking bee flight paths," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Vol. 5, VISAPP* (Funchal: INSTICC; SciTePress), 563–571.

Marstaller, J., Tausch, F., and Stock, S. (2019). "DeepBees - Building and scaling convolutional neuronal nets for fast and large-scale visual monitoring of bee hives," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (Seoul: IEEE), 271–278.

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

Meikle, W. G., and Holst, N. (2016). Application of continuous monitoring of honeybee colonies. *Springer Verlag* 46, 10–22. doi: 10.1007/s13592-014-0298-x

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. Protocols* 14, 2152–2176. doi: 10.1038/s41596-019-0176-0

Neumann, P., and Carreck, N. L. (2010). Honey bee colony losses. *J. Apic. Res.* 49, 1–6. doi: 10.3896/IBRA.1.49.1.01

Nystrom, N. A., Levine, M. J., Roskies, R. Z., and Scott, J. R. (2015). "Bridges: a uniquely flexible HPC resource for new communities and data analytics," in *Proceedings of the 2015 XSEDE Conference on Scientific Advancements Enabled by Enhanced Cyberinfrastructure - XSEDE '15* (St. Louis, MO: ACM Press), 1–8.

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., et al. (2019). Fast animal pose estimation using deep neural networks. *Nat. Methods* 16, 117–125. doi: 10.1038/s41592-018-0234-5

Redmon, J., and Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv e-prints*, arXiv:1804.02767.

Robie, A. A., Seagraves, K. M., Egnor, S. E. R., and Branson, K. (2017). Machine vision methods for analyzing social interactions. *J. Exp. Biol.* 220, 25–34. doi: 10.1242/jeb.142281

Rodriguez, I. F., Mégret, R., Acuña, E., Agosto-Rivera, J. L., and Giray, T. (2018a). "Recognition of pollen-bearing bees from video using convolutional neural network," in *IEEE Winter Conference on Applications of Computer Vision (WACV)* (Lake Tahoe, NV: IEEE), 314–322.

Rodriguez, I. F., Mégret, R., Egnor, R., Branson, K., Acuña, E., Agosto-Rivera, J. L., et al. (2018b). "Multiple animals tracking in video using part affinity fields," in *Workshop Visual observation and analysis of Vertebrate And Insect Behavior (VAIB) at International Conference on Pattern Recognition (ICPR)* (Beijing).

Romero-Ferrero, F., Bergomi, M. G., Hinz, R., Heras, F. J. H., and de Polavieja, G. G. (2018). idtracker.ai: Tracking all individuals in large collectives of unmarked animals. *CoRR, abs/1803.04351*. doi: 10.1101/280735

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR, abs/1409.1556*.

Sledevič, T. (2018). "The application of convolutional neural network for pollen bearing bee classification," in *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (Vilnius: IEEE), 1–4.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., et al. (2014). XSEDE: accelerating scientific discovery. *Comput. Sci. Eng.* 16, 62–74. doi: 10.1109/MCSE.2014.80

Wang, J., and Olson, E. (2016). "AprilTag 2: efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon: IEEE), 4193–4198.

Wario, F., Wild, B., Couvillon, M., Rojas, R., and Landgraf, T. (2015). Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees. *Front. Ecol. Evol.* 3:103. doi: 10.3389/fevo.2015.00103

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 Computer Science

METHODS
published: 10 February 2022
doi: 10.3389/fcomp.2022.777728

LABKIT: Labeling and Segmentation Toolkit for Big Image Data

Matthias Arzt[1,2], *Joran Deschamps*[1,2,3], *Christopher Schmied*[3], *Tobias Pietzsch*[1,2], *Deborah Schmidt*[1,2,4], *Pavel Tomancak*[1,2,5], *Robert Haase*[1,2,6] *and Florian Jug*[1,2,3]*

[1] Center for Systems Biology Dresden, Dresden, Germany, [2] Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany, [3] Fondazione Human Technopole, Milan, Italy, [4] Max Delbrück Center for Molecular Medicine, Berlin, Germany, [5] IT4Innovations, VŠB-Technical University of Ostrava, Ostrava, Czechia, [6] DFG Cluster of Excellence "Physics of Life", TU-Dresden, Dresden, Germany

We present LABKIT, a user-friendly Fiji plugin for the segmentation of microscopy image data. It offers easy to use manual and automated image segmentation routines that can be rapidly applied to single- and multi-channel images as well as to timelapse movies in 2D or 3D. LABKIT is specifically designed to work efficiently on big image data and enables users of consumer laptops to conveniently work with multiple-terabyte images. This efficiency is achieved by using ImgLib2 and BigDataViewer as well as a memory efficient and fast implementation of the random forest based pixel classification algorithm as the foundation of our software. Optionally we harness the power of graphics processing units (GPU) to gain additional runtime performance. LABKIT is easy to install on virtually all laptops and workstations. Additionally, LABKIT is compatible with high performance computing (HPC) clusters for distributed processing of big image data. The ability to use pixel classifiers trained in LABKIT via the ImageJ macro language enables our users to integrate this functionality as a processing step in automated image processing workflows. Finally, LABKIT comes with rich online resources such as tutorials and examples that will help users to familiarize themselves with available features and how to best use LABKIT in a number of practical real-world use-cases.

Keywords: segmentation, labeling, machine learning, random forest, Fiji, open-source

OPEN ACCESS

Edited by:
Marcello Pelillo,
Ca' Foscari University of Venice, Italy

Reviewed by:
Xinggang Wang,
Huazhong University of Science and Technology, China
Csaba Beleznai,
Austrian Institute of Technology (AIT), Austria

***Correspondence:**
Florian Jug
florian.jug@fht.org

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *15 September 2021*
Accepted: *13 January 2022*
Published: *10 February 2022*

Citation:
Arzt M, Deschamps J, Schmied C, Pietzsch T, Schmidt D, Tomancak P, Haase R and Jug F (2022) LABKIT: Labeling and Segmentation Toolkit for Big Image Data. Front. Comput. Sci. 4:777728. doi: 10.3389/fcomp.2022.777728

1. INTRODUCTION

In recent years, new and powerful microscopy and sample preparation techniques have emerged, such as light-sheet (Huisken et al., 2004), super-resolution microscopy (Hell and Wichmann, 1994; Gustafsson, 2000; Betzig et al., 2006; Hess et al., 2006; Rust et al., 2006), modern tissue clearing (Dodt et al., 2007; Hama et al., 2011), or serial section scanning electron microscopy (Denk and Horstmann, 2004; Knott et al., 2008) enabling researchers to observe biological tissues and their underlying cellular and molecular composition and dynamics in unprecedented details. To localize objects of interest and exploit such rich datasets quantitatively, scientists need to perform image segmentation, e.g., dividing all pixels in an image into foreground pixels (part of objects of interest) and background pixels.

The result of such a pixel classification is a binary mask, or a (multi-)label image if more than one foreground class is needed to discriminate different objects. Masks or label images enable downstream analysis that extract biologically meaningful semantic quantities, such as the number of objects in the data, morphological properties of these objects (shape, size, etc.), or

tracks of object movements over time. In most practical applications, image segmentation is not an easy task to solve. It is often rendered difficult by the sample's biological variability, imperfect imaging conditions (e.g., leading to noise, blur, or other distortions), or simply by the complicated three-dimensional shape of the objects of interest.

Current research in bio-image segmentation focuses primarily on developing new deep learning approaches, with more classical methods currently receiving little attention. Algorithms, such as StarDist (Schmidt et al., 2018), DenoiSeg (Buchholz et al., 2020), PatchPerPix (Mais et al., 2020), PlantSeg (Wolny et al., 2020), CellPose (Stringer et al., 2021), or EmbedSeg (Lalit et al., 2021) have continuously raised the state-of-the art and outperform classical methods in quality and accuracy of achieved automated segmentation. While these approaches are very powerful indeed, deep learning does require some expert knowledge, dedicated computational resources not everybody has access to, and typically large quantities of densely labeled ground-truth data to train on.

More classical approaches, on the other hand, can also yield results that enable the required analysis, while often remaining fast and easy to use on any laptop or workstation. Examples for such methods range from intensity thresholding and seeded watershed, to shallow machine learning approaches on manually chosen or designed features. One crucial property of shallow techniques, such as random forests (Breiman, 2001), is that they require orders of magnitude less ground-truth training data than deep learning based methods. Hence, multiple software tools pair them with user-friendly interfaces, e.g., CellProfiler (McQuin et al., 2018), Ilastik (Berg et al., 2019), QuPath (Bankhead et al., 2017), and Trainable Weka Segmentation (Arganda-Carreras et al., 2017). The latter specializes in random forest classification and is available within Fiji (Schindelin et al., 2012), a widely-used image analysis and processing platform based on ImageJ (Schneider et al., 2012) and ImageJ2 (Rueden et al., 2017). It is, regrettably, not capable of processing very large datasets due to its excessive demand for CPU memory, leaving the sizable Fiji community with a lack of user-friendly pixel classification or segmentation tools that can operate on large multi-dimensional data.

The required foundations for such a software tool have in recent years been built by the vibrant research software engineering community around Fiji and ImageJ2. Specifically, the problem of handling large multi-dimensional images has been addressed by a generic and powerful library called ImgLib2 (Pietzsch et al., 2012). Additionally, a fast, memory-efficient, and extensible image viewer, the BigDataViewer (Pietzsch et al., 2015), enables tool developers to create intuitive and fast data handling interfaces.

Here, we present an image labeling and segmentation tool called LABKIT. It combines the power of ImgLib2 and BigDataViewer with a new implementation of random forest pixel classification. LABKIT features a user-friendly interface allowing for rapid scribble labeling, training, and interactive curation of the segmented image. LABKIT also allows users to fully manually label pixels or voxels in the loaded images. It can be easily installed in Fiji, and directly called from its macro programming language. LABKIT additionally features GPU acceleration using CLIJ (Haase et al., 2020), and can be used on high performance computing (HPC) clusters thanks to a command-line interface.

2. IMAGE SEGMENTATION WITH LABKIT

LABKIT's user interface is built around the BigDataViewer (Pietzsch et al., 2015), which allows interactive exploration of image volumes of any size and dimension on consumer computing hardware (**Figures 1A,B**). Beyond the common BigDataViewer features, users have access to a set of simple drawing tools to manually paint or correct existing labels on image pixels in 2D and voxels in 3D. Importantly, the raw data is never modified by any such actions. Pixel and voxel labels are grouped by classes in individual layers (e.g., background, nucleus or organelle). Each class is represented by a modifiable color, and can be used to annotate different types of objects and structures of interest in the image.

Thanks to the intuitive interface design, users can efficiently segment their images by manually drawing dense labels on the entire image (**Figure 1C**). Labels that are generated with the drawing tools can directly be saved as images or exported to Fiji for downstream processing. Dense manual labelings of complete images or volumes created with LABKIT can be used to manually segment objects, as was done previously to mask particles in cryo-electron tomograms of *Chlamydomonas* (Jordan and Pigino, 2019).

However, this process is very time consuming and doesn't scale well to large data. LABKIT is therefore often used to densely and manually label a subset of the image data, which is then used as ground-truth for supervised deep learning approaches. Published examples include the generation of ground-truth training data for a mouse and a *Platyneris* dataset in order to segment cell nuclei with EmbedSeg (Lalit et al., 2021). LABKIT is also suggested as a tool of choice for ground-truth generation by other deep learning methods (Schmidt et al., 2018; Buchholz et al., 2020; Horlava et al., 2020). Still, manually generating sufficient amount of ground-truth training labels for existing deep learning methods remains a cumbersome and tedious task.

In order to create a high quality segmentation while maintaining low user input, LABKIT features a random forest (Breiman, 2001) based pixel classification algorithm, with all feature computations optimized for quick runtimes. Instead of annotating entire objects, a random forest is trained on a few pixel labeling per class only. These sparse manual labels, or scribbles (see **Figure 1D**, left), are directly drawn by users over the image. Naturally, scribbles must be drawn on pixels representative of each class. Once trained, the random forest classifier enables the generation of a segmentation (dense pixel classification, see **Figure 1D**). Two or more classes can be used to distinguish foreground objects from background pixels. **Figures 2A,B** showcase examples of a single foreground and background classes. If desired, out of focus objects can even be discarded, for example by making such pixels part of the background class (**Figure 2B**, arrowheads). For more complex

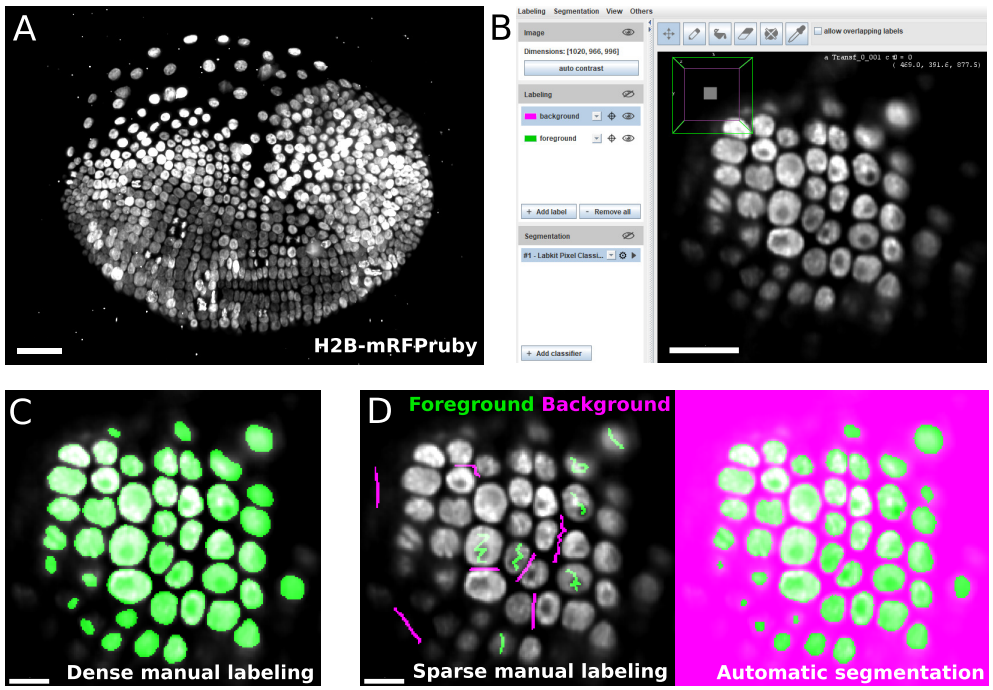


FIGURE 1 | LABKIT allows easy manual labeling and automatic segmentation of large image volumes: **(A)** Maximum intensity projection of a single time point from a ~1 TB timelapse of a developing *Parhyale* embryo imaged live with lightsheet microscopy. **(B)** LABKIT's user interface is based on BigDataViewer and allows visualizing and interacting with large volumes of image data. A slice of the developing *Parhyale* embryo is shown. **(C)** Users can label large datasets with dense manual annotations using LABKIT's drawing interface. **(D)** A core feature of LABKIT is the rapid segmentation of large image data using sparse manual labels (scribbles) combined with random forest pixel classification to automatically produce the final segmentation. Scale bars 100 μm **(A)**, 50 μm **(B)**, and 25 μm **(C,D)**.

segmentation tasks that need to discriminate various visible structures (e.g., nucleus vs. cytoplasm vs. background) or cell types (as in **Figure 2C**), two or more foreground classes can be used (**Figure 2D**).

As opposed to deep learning algorithms, random forests are typically trained in a matter of seconds. Drawing scribbles and computing the segmentation can therefore conveniently be iterated due to the efficient parallelization we have implemented, leading to live segmentation. Live results are computed and displayed only on the currently visualized image slice in BigDataViewer to increase the interactivity. Hence, the effect of additional scribbles (sparse labels) is instantly visible and users can stop once the automated output of the pixel classifier reaches sufficient quality. This iterative workflow makes working with LABKIT very efficient, even when truly large image data are being processed. BigDataViewer's bookmarking feature can additionally be used to quickly jump between previously defined image regions, thereby allowing validating the quality of the pixel classifier on multiple areas. Since we use ImgLib's caching infrastructure, all image blocks that have once been computed are kept in memory and switching between bookmarks or browsing between parts of a huge volume is fast and visually pleasing. Once sufficiently trained, the classifier can be saved for later use in interactive LABKIT sessions or in Fiji/ImageJ macros. The entire dataset can be directly segmented and the results saved to disk. Recently, sparse labeling combined with random forest pixel classification in LABKIT was used to segment mice epidermal cells (Bornes et al., 2021), as well as mRNA foci in neurons (Arshadi et al., 2021).

Once the image is fully segmented, the generated segmentation masks can be transferred to label layers and the drawing tools can now be used to curate them. The goal of curation is to resolve the remaining errors made by the trained pixel classifier, such as drawing missing parts, filling holes, erasing mislabeling and deleting spurious blobs (**Figure 3**). Label curation is performed until the curated segmentation is deemed satisfactory for downstream processing or analysis. LABKIT can also be used to curate segmentation results obtained by other methods that are not available within LABKIT, including deep learning based methods (Jain et al., 2020).

Automated segmentation with LABKIT and the possibility to quickly curate any automated segmentation result make LABKIT a powerful tool that can considerably shorten the time required to generate ground-truth data for training deep learning approaches. For example, we compared automatic and manual segmentation with LABKIT on a rather small subset of images (N=26, see one example in **Figure 4A**) made publicly available by the 218 Data Science Bowl (Caicedo et al., 2019). We segmented all images within 5 min by iterative scribbling and automated segmentation (see **Figure 4B**). While many images consisted of homogeneous nuclei and led to high quality results, images with heterogeneous nuclei resulted in segmentation errors (see

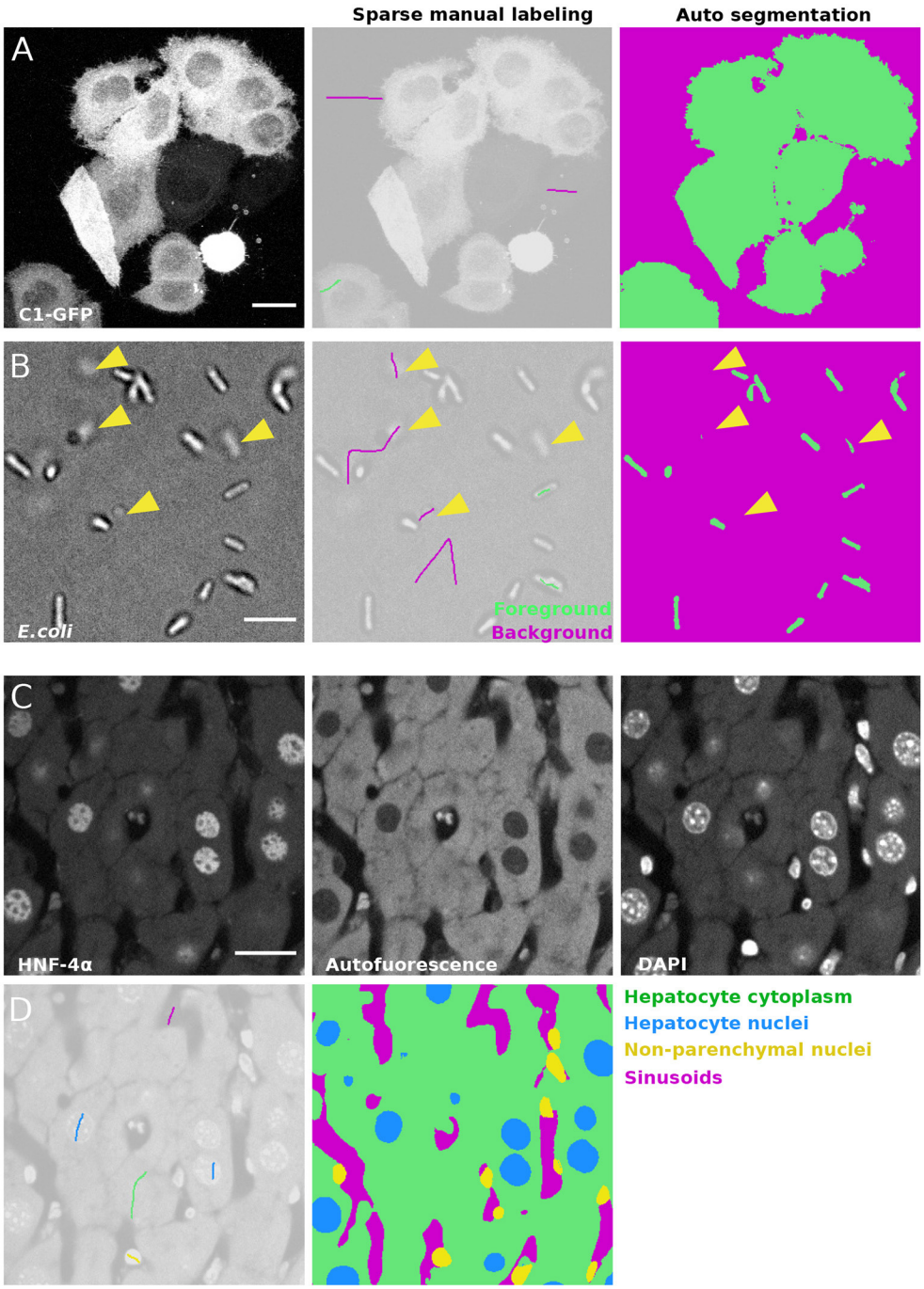


FIGURE 2 | Semantic segmentation of microscopy images with LABKIT's pixel classification: **(A)** Maximum intensity projection of a confocal stack showing HeLa cells expressing C1-GFP (left), next to the sparse labeling (scribbles, center) and resulting cell segmentation (right). **(B)** Bright field microscopy image of *E.coli*, sparse labeling discriminating cells and background and the resulting segmentation. Arrowheads show that segmentation of out-of-focus objects can be reduced by including pixels of such objects in the background class. **(C)** Fixed mouse liver tissue section stained with immunofluorescence and imaged in multiple channels with a spinning disk confocal microscope, showing Hepatocyte nuclei stained with antibody against HNF-4α a transcription factor expressed in hepatocytes, hepatocyte cytoplasm (autofluorescence) and all nuclei stained with DAPI. **(D)** Labeling and resulting segmentation of the liver tissue section shown in **(A)**, segmenting Hepatocyte cytoplasm (green), Hepatocyte nuclei (blue), nuclei of non-parenchymal cells (yellow) and sinusoids (magenta). Scale bars 20 μm **(A)**, **(C)**, and 5 μm **(B)**.

arrows in **Figure 4B**). Such errors include spurious instances that do not correlate with any object in the original image, instances that correspond to the fusion of multiple instances, instances with holes, or even instances that split in two. Such errors are obviously undesirable and negatively impact the overall average precision score (AP = 0.72, see Methods for the metrics

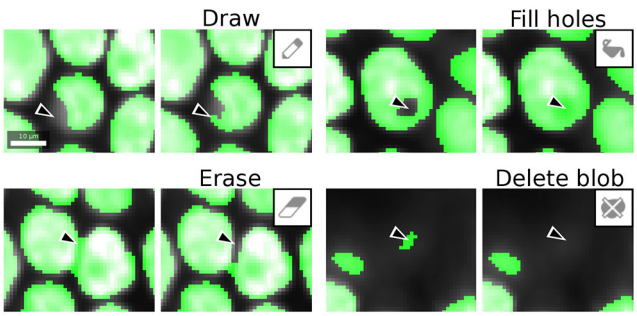


FIGURE 3 | LABKIT labeling tools used for curation: labels generated by manual dense labeling or automatic segmentation can be efficiently curated with drawing, filling, erasing, or deletion of entire objects. Scale bar 10 μm.

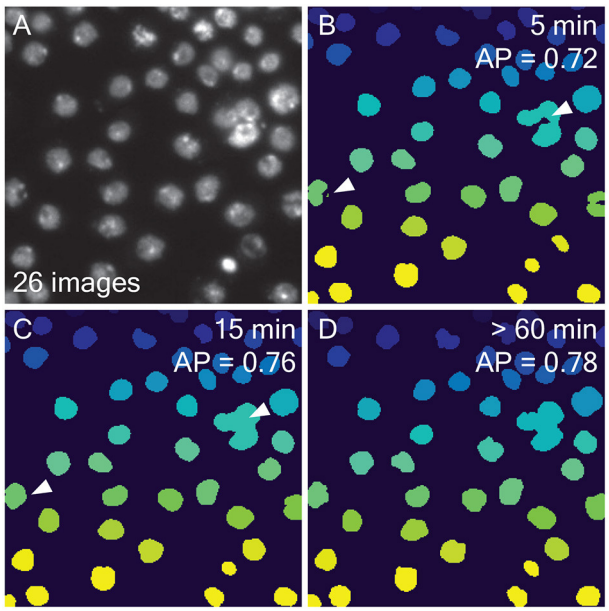


FIGURE 4 | Comparing automatic and manual ground-truth generation with LABKIT: **(A)** Fluorescence image of nuclei (out of 26 images) extracted from the 2018 Data Science Bowl (Caicedo et al., 2019). **(B)** Results from LABKIT automated segmentation of **(A)** after extracting connected components and giving each instance a unique pixel value. The arrows point to various segmentation errors. On the top right corner, the total time necessary to obtain the corresponding segmentation of all 26 images (including labeling) is indicated. Below the timing is the average precision (see Methods) as compared to a dense manual labeling performed by another observer. **(C)** Curation of **(B)** with same post-processing. The arrows point to the corrected errors mentioned in **(B)**. The timing information includes **(B)**. **(D)** Dense manual labeling of **(A)** and the same post-processing as in **(B)**. No scale bar was available for the images.

definition). As described above, all such segmentation errors can easily be corrected within LABKIT, either by adding sparse labels corresponding to typical areas with errors, done during the iterative process, or when they persist by manually curating the residual errors in the final automated results (**Figure 4C**). Curating all 26 images took an additional 10 min and raised the corresponding average precision to 0.76, a score very close to the inter-observer distance (AP = 0.78), as shown in **Figures 4C,D**. In contrast, manually segmenting all images required more than an hour (**Figure 4D**), which is four times longer than scribble-based pixel classification with LABKIT, followed by full curation of the results to obtain images of comparable quality.

Hence, whenever LABKIT automated segmentation is by itself not sufficient, manually curating the results yields ground-truth data that can be used to train a deep learning method, leading to higher segmentation quality with less labeling effort.

3. LABKIT PIXEL CLASSIFIER

LABKIT provides a pixel classification algorithm for automatic segmentation. The algorithm uses a random forest to classify each pixel independently into user-defined classes (e.g., foreground and background). Random Forests (Breiman, 2001) are widely used supervised machine learning methods, and as such must be trained on a given body of ground-truth labels (pre-classified example pixels). In LABKIT, the random forest classifier is trained on manually labeled pixels (scribbles), an approach similar to ilastik (Berg et al., 2019) or Trainable Weka Segmentation (Arganda-Carreras et al., 2017). As opposed to most implementations, LABKIT's classifier is specifically optimized to be able to handle very large image data.

In a first step, we compute a feature vector for each labeled pixel. This is achieved by applying a configurable set of filters to the given image or images. To this end LABKIT offers a set of image filters commonly used in image analysis, such as Gaussian, difference of Gaussians or Laplacian filters. Each selected filter creates an output image that emphasizes different features of a given input image. Filter responses for each pixel are then added to their feature vector. The final feature vectors of all labeled pixels are paired with their respective ground-truth classes, together constituting the training set. This data is then used to train the random forest, consisting of 100 decision trees, using the FastRF library (Supek, 2015).

After training, the random forest classifier can predict pixel classes directly from the feature vector of any given pixel. Hence, in a final step, we apply the random forest to the feature vectors of all pixels in the entire body of data, thereby effectively computing the desired semantic segmentation.

Since computing feature vectors and final random forest predictions consume by far the most computational resources, it was crucial to optimize their runtime. To this end, we process image chunks in parallel, with the chunked memory handling being supported by ImgLib2 (Pietzsch et al., 2012). Additionally, we implemented OpenCL kernels, allowing us to benefit from fast GPU computations (Haase et al., 2020).

4. LIMITATIONS OF THE PIXEL CLASSIFICATION

The simplicity of the pixel classification algorithm ensures efficiency, but also exposes it to certain limitations and potential

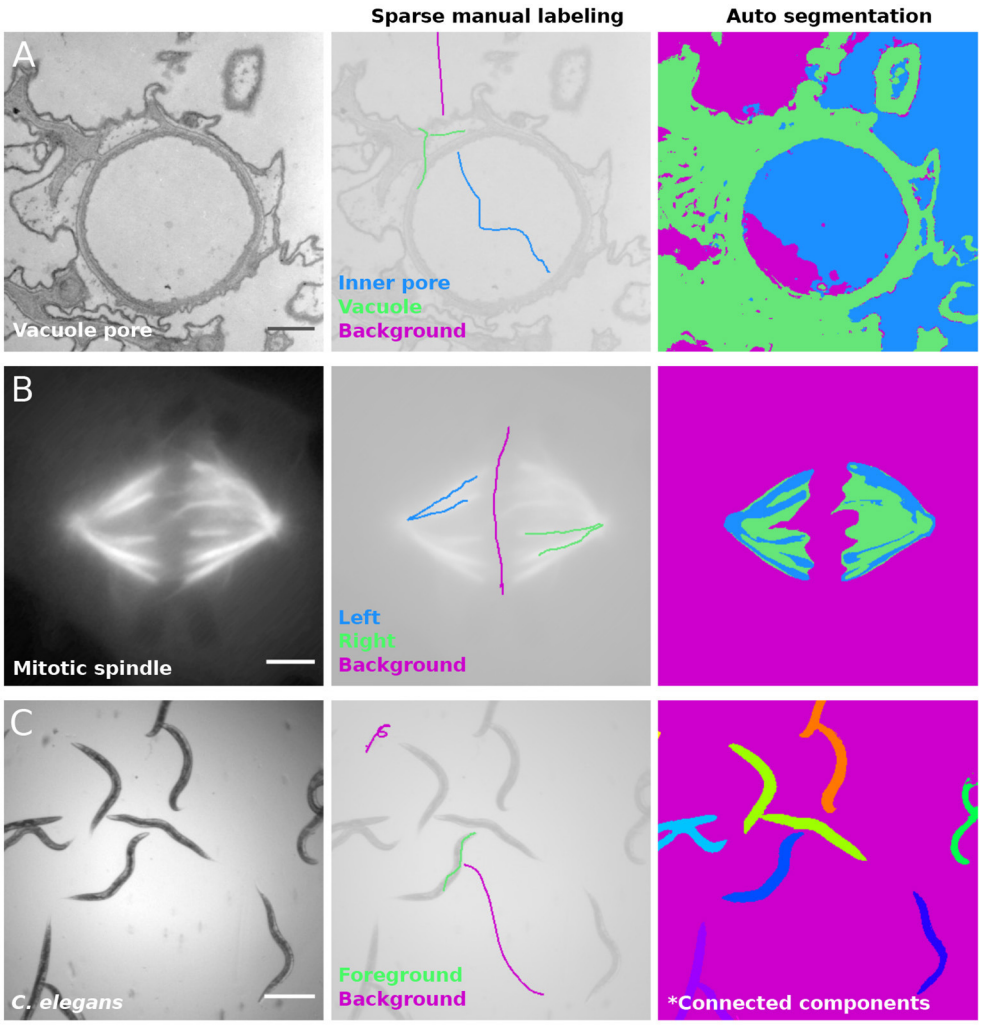


FIGURE 5 | Pixel classification failure modes: **(A)** Image of a contractile vacuole pore in *paramecium caudatum* acquired with transmission electron microscopy (left), in which inner pore (blue), cytoplasm (green) and background (magenta) were labeled with scribbles (center). LABKIT's classifier fails to distinguish background and inner pore classes (right). Image provided by Richard Allen (Cell Image Library, 38894). **(B)** Mitotic spindle of a Ptk2 cell expressing GFP-tubulin imaged in wide-field fluorescence (left). Left and right halves of the mitotic spindle are labeled in different classes (blue and green, respectively), while background is labeled in magenta (center). LABKIT's classifier is incapable of discriminating similar structures (right). Image provided by Sophie Dumont and Timothy J. Mitchison (Cell Image Library, 6568). **(C)** Bright field image of *C. elegans* (left), labeled with two classes (center): foreground (green) and background (magenta). After applying connected components analysis to the classification result, contiguous worms belong to the same object (right). Image provided by Fred Ausubel (Broad Bioimage Benchmark Collection, BBBC010). Scale bars **(A)** 500 nm, **(B)** 10 μm, and **(C)** 500 μm.

failure modes. This mainly comes from the fact that the filter kernels used to compute the feature vector have limited sizes. With the default settings in LABKIT, filters respond mostly to a 16x16 window (2D), meaning that decision about the class of a given pixel is based on filter responses in a small neighborhood. A direct consequence is illustrated in **Figure 5A**, where LABKIT was used to segment the image of a vacuole pore in *paramecium caudatum* (left panel) using three classes: inner pore (central panel, in blue), vacuole (in green) and background (in magenta). Because inner pore and background pixels have similar texture, the classifier cannot tell them apart and assigns background pixels to the inner pore class, and vice versa.

All filters used to calculate the feature vectors are designed to be translation and rotation invariant. Hence, the algorithm classifies two objects the same way regardless of their position or orientation in the image. While this is a sensible assumption for most applications on microscopy data, users should certainly be aware of this. One potential problem is showcased in **Figure 5B**, which shows a mitotic spindle in fluorescence microscopy (left) and an attempt at assigning each half of the mitotic spindle to a different class (central panel). Although both sides of the mitotic spindle are spatially distinct and in different orientations, the classifier fails at discriminating them (**Figure 5B**, right).

Furthermore, contiguous objects cannot be separated by the random forest classifier. This is illustrated in **Figure 5C**, which

TABLE 1 | Benchmarking computation speed while segmenting a large biological image on various hardware: the experiment was performed on a laptop with and without GPU acceleration, and on different numbers of CPU and GPU cluster nodes.

Hardware		GPU	Run time	Speed-up	Throughput in gigapixel
	Laptop	No	4 h 23 min 00 s	1	3.05 / h = 0.05 / min
	Laptop	Yes	35 min 12 s	7.5	0.38 / min
1	CPU cluster node	No	1 h 08 min 10 s	1	0.20 / min
10	CPU cluster nodes	No	6 min 15 s	10.9	2.14 / min
50	CPU cluster nodes	No	1 min 35 s	43.1	8.45 / min
1	GPU cluster node	Yes (2)	8 min 23 s	1	1.60 / min
10	GPU cluster nodes	Yes (2)	1 min 03 s	7.9	12.74 / min

In each category, the speed-up is calculated in comparison to the slower entry. Numbers in between parenthesis in the GPU column indicate the number of GPU per cluster node.

shows *C. elegans* worms imaged in bright-field microscopy (left panel). While the classifier can correctly distinguish the worms body from the background, a connected component analysis applied to the classification result (**Figure 5C**, right) leads to multiple worms being fused within the same connected component. In order to obtain instance segmentation from such images, manual curation or post-processing, such as watershed, is necessary to separate the connected objects into different instances.

Finally it is also important to know that trained random forest classifier cannot easily be trained on sets of very diverse images. Deep learning approaches such as Cellpose (Stringer et al., 2021), in contrast, show much greater potential to generalize well even when trained on a large and diverse body of microscopy data.

Nonetheless, LABKIT can be used to segment a wide range of images fast and at high quality. This is true as long as objects are visibly separated from one another and can be distinguished by the filter responses LABKIT computes per pixel.

5. SOFTWARE AND WORKFLOW INTEGRATION

LABKIT's automatic segmentation is not limited to the dataset it was trained on. Because the trained classifier can be saved for later use, it can be applied to similar new images. While ensuring reproducibility of the results, it also helps maintaining consistency in the image segmentation. Manually loading both images and trained classifier in LABKIT for multiple sets of images is a repetitive task ill-suited for an automated workflow. Therefore, to simplify the integration into existing workflows in Fiji, LABKIT can be easily called from the ImageJ macro language. For instance, a simple macro script can open multiple datasets and segment each of them using a trained classifier.

Image segmentation can be further accelerated by running the process on GPUs thanks to CLIJ (Haase et al., 2020). Once CLIJ is properly set up, GPU acceleration is available for LABKIT in both graphical interface and macro commands. GPU processing is particularly beneficial in the case of large images, for which it allows shortening the lengthy segmentation tasks. Performing GPU-accelerated segmentation in LABKIT is a matter of activating a checkbox, and does not present additional complexity to users.

Some images, however, are far too large to be processed on a consumer machine in a reasonable amount of time, if they can be stored at all on such a computer. For such data, modern workflows resort to the use of HPC clusters, which are purposely built for high computing performances with large available memory. LABKIT offers a command line tool (Arzt, 2021a) allowing advanced users to segment images on HPC clusters.

The capability of extending LABKIT and re-using its components is illustrated by integration with the commercial Imaris software (Oxford Instruments, UK) via the recently released ImgLib2-Imaris compatibility bridge. In this context, LABKIT operates directly on datasets that are transparently shared (without duplication) between Imaris and ImgLib2 (Pietzsch et al., 2012). These datasets can be arbitrarily large, as both Imaris and ImgLib2 implement sophisticated caching schemes. In the same fashion, output segmentation masks are transparently shared with the running Imaris application, making additional file import/export steps unnecessary. Importantly, this functionality can also be triggered and controlled directly from Imaris to integrate it into streamlined object segmentation workflows.

6. PERFORMANCE OF LABKIT

In order to process large images on consumer computers, software packages must be able to load the data in memory, process it and save the results, all within the constraints of the machine. In LABKIT, this is achieved by reading only the portions of the image that are displayed to the user, thanks to the use of the HDF5 format (Folk et al., 2011) and the BigDataViewer (Pietzsch et al., 2015). The image is further processed in chunks using ImgLib2 (Pietzsch et al., 2012). As a result, LABKIT is capable of processing arbitrarily large

images and is compatible with GPU acceleration and distributed computation on HPC clusters.

To illustrate this, we segmented a 13.4 gigapixel image (482 x 935 x 495 x 60 pixels, 25 GB) on a single laptop computer, with and without GPU, and with different nodes of an HPC cluster (see **Table 1**). The image was extracted and 2x down-sampled from the *Fluo-N3DL-TRIF* dataset made available for the Cell Tracking Challenge (Maška et al., 2014; Ulman et al., 2017; Jain et al., 2020) benchmark competition. Running the segmentation on the laptop using GPU acceleration sped up the computation by 7.5 fold, illustrating the benefit of harnessing GPU power for processing large images. While running computation on an HPC cluster comes with overhead, increasing the number of CPU nodes shortens the computation dramatically, reaching a 40-fold improvement from 1 CPU node to 50. Finally, GPU nodes on an HPC allow for more parallelization of the computation and therefore even higher computational speed-up on the segmentation task, with 10 GPU nodes processing the data in slightly over a minute.

Furthermore, we trained and optimized a classifier on the *Fluo-N3DL-TRIF* dataset (original sampling), the largest dataset of the Cell Tracking Challenge (training dataset of size 320 GB, evaluation dataset of size 467 GB), and submitted it for evaluation against undisclosed ground-truth. The segmentation of both training and evaluation datasets was performed on an HPC cluster. LABKIT pixel classification ranked as the highest performing segmentation method on this dataset for all three evaluation metrics (OP_{CSB}, *SEG* and *DET*) (CTC, 2021). More specifically, LABKIT segmentation obtained the following scores: $OP_{CSB} = 0.895$ (0.886 for the second highest scoring entry), $SEG = 0.793$ (0.776) and $DET = 0.997$ (0.997), performing better than the other entries, including classical (bandpass segmentation) or deep learning (convolution neural network) algorithms. As opposed to the deep learning algorithm to which it was compared, Labkit only used a few hundred pixels as ground-truth, distributed throughout a small fraction of the training dataset (7 frames). Finally, LABKIT's classifier was simply trained through the LABKIT graphical interface, illustrating its ease of use.

7. DISCUSSION AND CONCLUSION

LABKIT is a labeling software tool designed to be intuitive and simple to use. It features a robust pixel classification algorithm aimed at segmenting images between multiple classes with very little manual labeling required. Similar to other tools of the BigDataViewer family (Pietzsch et al., 2015; Wolff et al., 2018; Hörl et al., 2019; Tischer et al., 2020), it integrates seamlessly into the SciJava and Fiji ecosystem. It can be easily installed through Fiji and incorporated into established workflows using ImageJ's macro language. The results of LABKIT's segmentation can be further analyzed in Fiji or exported to other software platforms, such as CellProfiler (McQuin et al., 2018), QuPath (Bankhead et al., 2017), or Ilastik (Berg et al., 2019).

Manual labeling, in both 2D and 3D, is also made easy by LABKIT. Other alternatives exist, among which QuPath (Bankhead et al., 2017) (2D), napari (napari contributors, 2019) or Paintera (Leite et al., 2021). In particular, Paintera is specifically tailored to 3D labeling of crowded environment, but at the cost of a steeper learning curve.

LABKIT is compatible with a wide range of image formats since image data can be loaded directly from Fiji using Bio-Formats (Linkert et al., 2010). Nonetheless, in order to fully benefit from LABKIT optimizations for large images, users must first convert their terabyte-sized images to a file format allowing high-speed access to arbitrary located sub-regions of the image. This strategy is also employed by other software, with the example of Ilastik (Berg et al., 2019). One such format is HDF5 (Folk et al., 2011), and LABKIT uses in particular the BigDataViewer HDF5+XML variant. In Fiji, images can easily be saved in this format using BigStitcher (Hörl et al., 2019) or Multiview-Reconstruction (Preibisch et al., 2014; Icha et al., 2016).

In the Cell Tracking Challenge (Ulman et al., 2017; CTC, 2021), LABKIT segmentation outperformed other entries on a particular dataset, one being a deep learning approach. This method was designed as part of a cell segmentation and tracking pipeline on various images, and it is likely that recent and specialized deep learning segmentation algorithms, such as StarDist (Schmidt et al., 2018) or CellPose (Stringer et al., 2021), would perform overall better (Baltissen et al., 2018; Morone et al., 2020). Yet, the full potential of deep learning algorithms is only reached when a sufficient amount of ground-truth data is available, which is too frequently the limiting factor. Generating ground-truth data for a deep learning method is a tedious endeavor without the insurance of a perfect segmentation result. A safer strategy is therefore to first try shallow learning for segmentation tasks, before even thinking of moving to deep learning algorithms. In cases where higher segmentation quality is truly necessary, curated results from shallow learning can be used to generate the massive amount of ground-truth required to train a deep learning algorithm. As seen previously, LABKIT is useful in all these scenarios since it can be used to manually generate ground-truth annotations or to segment the images with shallow learning before curating the results in order to use them as ground-truth for other learning-based algorithms (see **Figure 6**).

In the future, we intend to extend LABKIT's functionalities to improve manual and automated segmentation. For instance, we will add a magic wand tool to select, fill, fuse or delete labels based on the pixel classification. Furthermore, we aim to add new segmentation algorithms, such as the deep learning algorithm DenoiSeg (Buchholz et al., 2020) already available in Fiji. In recent years, novel interactive deep learning approaches have also been shown to reduce the need for large amounts of densely labeled ground-truth data. In general, these approaches combine deep learning with interactive user guidance, for instance clicks on the extreme points of objects (Maninis et al., 2018), inside-outside guidance (Zhang et al., 2020), clicks within objects and boundaries that are iteratively refined (Luo et al., 2021) and a combination of clicks and squiggles inside objects (Alemi

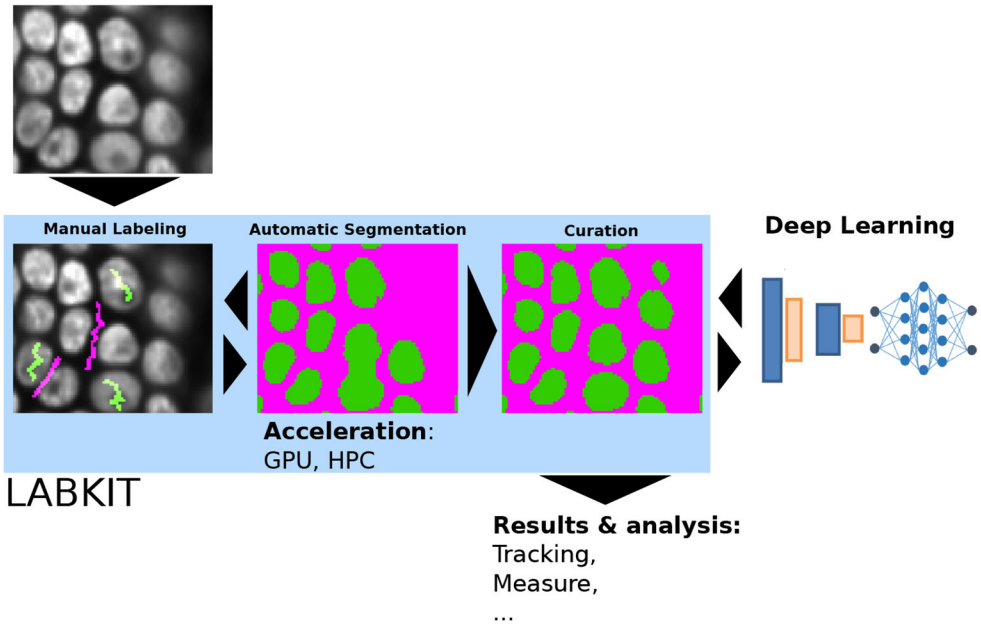


FIGURE 6 | LABKIT's iterative and interactive segmentation used for ground-truth generation: manual labeling, automatic segmentation and curation in LABKIT enable easy and rapid image segmentation, whose results can be further processed or used as ground-truth for deep-learning classifiers.

Koohbanani et al., 2020). However, these approaches are not in widespread use in bio-image analysis and for the most part implemented in Python. LABKIT could potentially serve as an easy-to-use platform for such methods by implementing their labeling strategies in the user interface and interfacing their framework with Java, thereby making them widely accessible to the biomedical community. LABKIT source code is open source and can be found online (Arzt, 2021b), together with its command-line interface (Arzt, 2021a), tutorials and documentation (Arzt, 2021c).

8. METHODS

8.1. Timing Instance Segmentation Generation

The dataset consisted of all 256 x 256 images (N = 26) in the test sample of StarDist (Schmidt et al., 2018), originally published as part of the 2018 Data Science Bowl (Caicedo et al., 2019) (subset of *stage1_train*, accession number BBBC038, Broad Bioimage Benchmark Collection). The images were loaded in LABKIT as a stack and sparsely labeled (scribbles). A classifier was then trained with the default filter settings: "original image," "Gaussian blur," "difference of Gaussians," "Gaussian gradient magnitude," "Laplacian of Gaussian," and "Hessian eigenvalues," with sigma values: 1, 2, 4, and 8. The results were saved and then manually curated using the brush and eraser tools. Finally, the same original image stack was densely manually labeled afresh. The total time required to process all images was measured using a chronometer for i) LABKIT automated segmentation, including the sparse manual labeling, ii) the previous step followed by a curation step and iii) dense manual labeling. In order to evaluate the segmented images, connected components were computed (4-connectivity) and given unique pixel values (instance segmentation). Quality metrics scores were calculated as the average precision with threshold 0.5 as defined in StarDist (Schmidt et al., 2018). We used dense manual labeling performed by another observer as reference images, and computed the metrics score for the results obtained in i), ii), and iii). The average metrics over the images were calculated as a weighted average of each individual image, where the weights were the number of instances in the reference image.

8.2. Speed Benchmark

The dataset was downloaded from the Cell Tracking Challenge (Ulman et al., 2017) website, and consisted of the first training dataset of the *Fluo-N3DL-TRIF* example. The dataset was down-sampled by a factor 2 in order to reduce its size and simplify the benchmarking. The dataset was then saved in the BigDataViewer XML+HDF5 format using BigStitcher (Hörl et al., 2019). LABKIT was used to draw a few scribbles on both background and nuclei areas, and to train a random forest classifier using the default settings. The trained model was then saved. The LABKIT command line tool was used to run the benchmark experiment on a Dell XPS 15 laptop (32 MB RAM, Intel Core i7-6700HQ CPU with 8 cores, GeForce GTX 960M GPU) and on an HPC cluster, with both CPU (256 GB RAM, Intel Xeon CPU E5-2680 v3 with 2.5 GHz and 24 cores) and GPU (512 GB RAM, Intel Xeon CPU E5-2698 v4 with 2.2 GHz and 40 cores, with two GeForce GTX 1080 GPUs) nodes. The segmentation results on the HPC were saved in the N5 (Saalfeld, 2017) format to maximize writing speed. Benchmarking included read/write of image data form disc, optional data transfer to the GPU, computation of feature images and classification all together.

8.3. Cell Tracking Challenge

As in the speed benchmark sample, all *Fluo-N3DL-TRIF* datasets (training and evaluation) were converted to BigDataViewer XML+HDF5 format using the BigStitcher Fiji plugin. This time, however, no down-sampling was applied to the images. For training, only frames 0, 1, 10, 20, 40, 50 and 59 from sequence "01" of the training dataset were used. A few hundred pixels were labeled as foreground and background. Only nuclei's central pixels were labeled as foreground in order to force the classification algorithm to return segments of smaller size than the actual nuclei. Thus, segmented nuclei are unlikely to touch and segmentation errors are minimized. We used the following filters to train the random forest classifier: "original image," "Gaussian blur," "Laplacian of Gaussian," and "Hessian eigenvalues," with sigma values 1, 2, 4, 8, and 16. The filters can be set in LABKIT's interface through the parameters menu of the classifier. The trained classifier was saved and the evaluation dataset was segmented using the LABKIT command line tool on an HPC. Since the output of the pixel classification is a binary mask, we performed a connected component analysis to assign unique pixel values to the individual segments. Finally, we dilated the segments to match the size of the nuclei. The dilation was done in three steps: the first two steps with a three-dimensional 6-neighborhood dilation kernel, then with a 3 x 3 x 3 pixel cube kernel. The combination of dilation kernels was chosen as to optimize the SEG score on the training dataset. All metrics scores were computed by the Cell Tracking Challenge platform.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://cellimagelibrary.org/images/38894, accession number CIL:38894, Cell Image Library, http://cellimagelibrary.org/images/6568, accession number CIL:6568, Cell Image Library, https://bbbc.broadinstitute.org/BBBC010, accession number BBBC010, Broad Bioimage Benchmark Collection, https://bbbc.broadinstitute.org/BBBC038, accession number BBBC038, Broad Bioimage Benchmark Collection and http://celltrackingchallenge.net/3d-datasets/, Fluo-N3DL-TRIF dataset, Cell Tracking Challenge.

AUTHOR CONTRIBUTIONS

FJ, MA, TP, PT, and DS designed the project. MA implemented the software with help from TP, DS, and RH. MA and JD performed experiments. MA, JD, CS, and FJ wrote the manuscript with inputs from all authors.

FUNDING

Funding was provided from the Max-Planck Society under project code M.IF.A.MOZG8106, the core budget of the Max-Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG), the Human Technopole, and the BMBF under codes 031L0102 (de.NBI) and 01IS18026C (ScaDS2), as well as by the Deutsche Forschungsgemeinschaft (DFG) under code JU3110/1-1 (FiSS) and TO563/8-1 (FiSS). PT was supported by the European Regional Development Fund in the IT4Innovations national supercomputing center, project number CZ.02.1.01/0.0/0.0/16_013/0001791 within the Program Research, Development and Education. RH acknowledges support by the DFG under Germany's Excellence Strategy–EXC2068-Cluster of Excellence Physics of Life of TU Dresden.

ACKNOWLEDGMENTS

We thank Anne Wuttke (Zerial lab, MPI-CBG), Sascha Kuhn (Nadler lab, MPI-CBG), Maria Luisa Romero Romero (Toth-Petroczy lab, MPI-CBG), Akanksha Jain & Anastasios (Tassos) Pavlopoulos (Tomancak, MPI-CBG) for sharing the experimental data. We also want to thank the Scientific Computing Facility at MPI-CBG for giving us access to HPC infrastructure.

REFERENCES

Alemi Koohbanani, N., Jahanifar, M., Zamani Tajadin, N., and Rajpoot, N. (2020). Nuclick: a deep learning framework for interactive segmentation of microscopic images. *Med. Image Anal.* 65:101771. doi: 10.1016/j.media.2020.101771

Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K. W., Schindelin, J., Cardona, A., et al. (2017). Trainable weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* 33, 2424–2426. doi: 10.1093/bioinformatics/btx180

Arshadi, C., Günther, U., Eddison, M., Harrington, K. I. S., and Ferreira, T. A. (2021). SNT: a unifying toolbox for quantification of neuronal anatomy. *Nat. Methods* 18, 374–377. doi: 10.1038/s41592-021-01105-7

Arzt, M. (2021a). Available online at: https://github.com/juglab/labkit-command-line (accessed December 09, 2021).

Arzt, M. (2021b). Available online at: https://github.com/juglab/labkit-ui (accessed December 09, 2021).

Arzt, M. (2021c). Available online at: https://imagej.net/plugins/labkit (accessed December 09, 2021).

Baltissen, D., Wollmann, T., Gunkel, M., Chung, I., Erfle, H., Rippe, K., et al. (2018). "Comparison of segmentation methods for tissue microscopy images of glioblastoma cells," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (Washington, DC: IEEE), 396–399.

Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., et al. (2017). Qupath: Open source software for digital pathology image analysis. *Sci. Rep.* 7, 1–7. doi: 10.1038/s41598-017-17204-5

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., et al. (2019). Ilastik: interactive machine learning for (bio) image analysis. *Nat. Methods* 16, 1226–1232. doi: 10.1038/s41592-019-0582-9

Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., et al. (2006). Imaging intracellular fluorescent proteins at nanometer resolution. *Science* 313, 1642–1645. doi: 10.1126/science.1127344

Bornes, L., Windoffer, R., Leube, R. E., Morgner, J., and van Rheenen, J. (2021). Scratch-induced partial skin wounds re-epithelialize by sheets of independently migrating keratinocytes. *Life Sci. Alliance* 4, e202000765. doi: 10.26508/lsa.202000765

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Buchholz, T.-O., Prakash, M., Krull, A., and Jug, F. (2020). DenoiSeg: joint denoising and segmentation. arXiv:2005.02987 [cs]. *arXiv: 2005.02987*. doi: 10.1007/978-3-030-66415-2_21

Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., et al. (2019). Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nat. Methods* 16, 1247–1253. doi: 10.1038/s41592-019-0612-7

CTC (2021). Available online at: http://celltrackingchallenge.net/latest-csb-results (accessed December 09, 2021).

Denk, W., and Horstmann, H. (2004). Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biol*. 2:e329. doi: 10.1371/journal.pbio.0020329

Dodt, H.-U., Leischner, U., Schierloh, A., Jährling, N., Mauch, C. P., Deininger, K., et al. (2007). Ultramicroscopy: three-dimensional visualization of neuronal networks in the whole mouse brain. *Nat. Methods* 4, 331–336. doi: 10.1038/nmeth1036

Folk, M., Heber, G., Koziol, Q., Pourmal, E., and Robinson, D. (2011). "An overview of the hdf5 technology suite and its applications," in *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases* (Uppsala), 36–47.

Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *J. Microsc*. 198, 82–87. doi: 10.1046/j.1365-2818.2000.00710.x

Haase, R., Royer, L. A., Steinbach, P., Schmidt, D., Dibrov, A., Schmidt, U., et al. (2020). Clij: Gpu-accelerated image processing for everyone. *Nat. Methods* 17, 5–6. doi: 10.1038/s41592-019-0650-1

Hama, H., Kurokawa, H., Kawano, H., Ando, R., Shimogori, T., Noda, H., et al. (2011). Scale: a chemical approach for fluorescence imaging and reconstruction of transparent mouse brain. *Nat. Neurosci*. 14, 1481–1488. doi: 10.1038/nn.2928

Hell, S. W., and Wichmann, J. (1994). Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Opt. Lett*. 19, 780–782. doi: 10.1364/OL.19.000780

Hess, S. T., Girirajan, T. P., and Mason, M. D. (2006). Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophys. J*. 91, 4258–4272. doi: 10.1529/biophysj.106.091116

Hörl, D., Rusak, F. R., Preusser, F., Tillberg, P., Randel, N., Chhetri, R. K., et al. (2019). Bigstitcher: reconstructing high-resolution image datasets of cleared and expanded samples. *Nat. Methods* 16, 870–874. doi: 10.1038/s41592-019-0501-0

Horlava, N., Mironenko, A., Niehaus, S., Wagner, S., Roeder, I., and Scherf, N. (2020). A comparative study of semi- and self-supervised semantic segmentation of biomedical microscopy data. arXiv:2011.08076 [cs, stat]. *arXiv: 2011.08076*.

Huisken, J., Swoger, J., Del Bene, F., Wittbrodt, J., and Stelzer, E. H. (2004). Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science* 305, 1007–1009. doi: 10.1126/science.1100035

Icha, J., Schmied, C., Sidhaye, J., Tomancak, P., Preibisch, S., and Norden, C. (2016). Using light sheet fluorescence microscopy to image zebrafish eye development. *J. Vis. Exp*. 110:e53966. doi: 10.3791/53966

Jain, A., Ulman, V., Mukherjee, A., Prakash, M., Cuenca, M. B., Pimpale, L. G., et al. (2020). Regionalized tissue fluidization is required for epithelial gap closure during insect gastrulation. *Nat. Commun*. 11, 5604. doi: 10.1038/s41467-020-19356-x

Jordan, M. A., and Pigino, G. (2019). "Chapter 9-*In situ* cryo-electron tomography and subtomogram averaging of intraflagellar transport trains," in *Methods in Cell Biology, volume 152 of Three-Dimensional Electron Microscopy*, eds T. Müller-Reichert and G. Pigino (Cambridge, MA: Academic Press), 179–195.

Knott, G., Marchman, H., Wall, D., and Lich, B. (2008). Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. *J. Neurosci*. 28, 2959–2964. doi: 10.1523/JNEUROSCI.3189-07.2008

Lalit, M., Tomancak, P., and Jug, F. (2021). Embedding-based Instance Segmentation in Microscopy. arXiv:2101.10033 [cs, eess]. *arXiv: 2101.10033*.

Leite, V., Saalfeld, S., Hanslovsky, P., Hulbert, C., Funke, J., Pietzsch, T., et al. (2021). *Paintera*. Zenodo. doi: 10.5281/zenodo.3351562

Linkert, M., Rueden, C. T., Allan, C., Burel, J.-M., Moore, W., Patterson, A., et al. (2010). Metadata matters: access to image data in the real world. *J. Cell Biol*. 189, 777–782. doi: 10.1083/jcb.201004104

Luo, X., Wang, G., Song, T., Zhang, J., Aertsen, M., Deprest, J., et al. (2021). Mideepseg: Minimally interactive segmentation of unseen objects from medical images using deep learning. *Med. Image Anal*. 72:102102. doi: 10.1016/j.media.2021.102102

Mais, L., Hirsch, P., and Kainmueller, D. (2020). "Patchperpix for instance segmentation," in Computer *Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16* (Glasgow: Springer), 288–304.

Maninis, K.-K., Caelles, S., Pont-Tuset, J., and Van Gool, L. (2018). "Deep extreme cut: From extreme points to object segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City: IEEE), 616–625.

Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., et al. (2014). A benchmark for comparison of cell tracking algorithms. *Bioinformatics* 30, 1609–1617. doi: 10.1093/bioinformatics/btu080

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B. A., Karhohs, K. W., et al. (2018). Cellprofiler 3.0: Next-generation image processing for biology. *PLoS Biol*. 16:e2005970. doi: 10.1371/journal.pbio.2005970

Morone, D., Marazza, A., Bergmann, T. J., and Molinari, M. (2020). Deep learning approach for quantification of organelles and misfolded polypeptide delivery within degradative compartments. *Mol. Biol.Cell*. 31, 1512–1524. doi: 10.1091/mbc.E20-04-0269

Napari Contributors (2019). *napari: A Multi-Dimensional Image Viewer For Python*. Zenodo. doi: 10.5281/zenodo.3555620

Pietzsch, T., Preibisch, S., Tomančák, P., and Saalfeld, S. (2012). Imglib2–generic image processing in java. *Bioinformatics* 28, 3009–3011. doi: 10.1093/bioinformatics/bts543

Pietzsch, T., Saalfeld, S., Preibisch, S., and Tomancak, P. (2015). Bigdataviewer: visualization and processing for large image data sets. *Nat. Methods* 12, 481–483. doi: 10.1038/nmeth.3392

Preibisch, S., Amat, F., Stamataki, E., Sarov, M., Singer, R. H., Myers, E., et al. (2014). Efficient Bayesian-based multiview deconvolution. *Nat. Methods* 11, 645–648. doi: 10.1038/nmeth.2929

Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T., et al. (2017). Imagej2: imagej for the next generation of scientific image data. *BMC Bioinformatics* 18:529. doi: 10.1186/s12859-017-1934-z

Rust, M. J., Bates, M., and Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm). *Nat. Methods* 3, 793–796. doi: 10.1038/nmeth929

Saalfeld, S. (2017). Available online at: https://github.com/saalfeldlab/n5 (accessed December 17, 2021).

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat Methods*. 9, 676–682. doi: 10.1038/nmeth.2019

Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. (2018). "Cell detection with star-convex polygons," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Berlin; Heidelberg: Springer), 265–273.

Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). Nih image to imagej: 25 years of image analysis. *Nat. Methods* 9, 671–675. doi: 10.1038/nmeth.2089

Stringer, C., Wang, T., Michaelos, M., and Pachitariu, M. (2021). Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* 18, 100–106. doi: 10.1038/s41592-020-01018-x

Supek, F. (2015). Available online at: https://code.google.com/archive/p/fast-random-forest/ (accessed December 09, 2021).

Tischer, C., Ravindran, A., Reither, S., Pepperkok, R., and Norlin, N. (2020). Bigdataprocessor2: a free and open-source fiji plugin for inspection and processing of tb sized image data. *bioRxiv*. doi: 10.1101/2020.09.23.244095

Ulman, V., Maška, M., Magnusson, K. E., Ronneberger, O., Haubold, C., Harder, N., et al. (2017). An objective comparison of cell-tracking algorithms. *Nat. Methods* 14, 1141–1152. doi: 10.1038/nmeth.4473

Wolff, C., Tinevez, J.-Y., Pietzsch, T., Stamataki, E., Harich, B., Guignard, L., et al. (2018). Multi-view light-sheet imaging and tracking with the mamut software reveals the cell lineage of a direct developing arthropod limb. *Elif*. 7:e34410. doi: 10.7554/eLife.34410

Wolny, A., Cerrone, L., Vijayan, A., Tofanelli, R., Barro, A. V., Louveaux, M., et al. (2020). Accurate and versatile 3d segmentation of plant tissues at cellular resolution. *Elife* 9:e57613. doi: 10.7554/eLife.57613

Zhang, S., Liew, J. H., Wei, Y., Wei, S., and Zhao, Y. (2020). "Interactive object segmentation with inside-outside guidance," in *2020 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), 12231–12241.

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 Computer Science

METHODS
published: 15 February 2022
doi: 10.3389/fcomp.2022.777615

μMatch: 3D Shape Correspondence for Biological Image Data

James Klatzow[1], *Giovanni Dalmasso*[2], *Neus Martínez-Abadías*[2,3], *James Sharpe*[2,4] *and Virginie Uhlmann*[1]*

[1] European Bioinformatics Institute (EMBL-EBI), European Molecular Biology Laboratory (EMBL), Cambridge, United Kingdom, [2] European Molecular Biology Laboratory Barcelona (EMBL), Barcelona, Spain, [3] Research Group in Biological Anthropology (GREAB), Department of Evolutionary Biology, Ecology, and Environmental Sciences (BEECA), Universitat de Barcelona, Barcelona, Spain, [4] Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain

Modern microscopy technologies allow imaging biological objects in 3D over a wide range of spatial and temporal scales, opening the way for a quantitative assessment of morphology. However, establishing a correspondence between objects to be compared, a first necessary step of most shape analysis workflows, remains challenging for soft-tissue objects without striking features allowing them to be landmarked. To address this issue, we introduce the μMatch 3D shape correspondence pipeline. μMatch implements a state-of-the-art correspondence algorithm initially developed for computer graphics and packages it in a streamlined pipeline including tools to carry out all steps from input data pre-processing to classical shape analysis routines. Importantly, μMatch does not require any landmarks on the object surface and establishes correspondence in a fully automated manner. Our open-source method is implemented in Python and can be used to process collections of objects described as triangular meshes. We quantitatively assess the validity of μMatch relying on a well-known benchmark dataset and further demonstrate its reliability by reproducing published results previously obtained through manual landmarking.

Keywords: bioimage analysis, shape quantification, correspondence, alignment, computational morphometry

OPEN ACCESS

Edited by:
Horst Bischof,
Graz University of Technology, Austria

Reviewed by:
Davide Boscaini,
Bruno Kessler Foundation (FBK), Italy
Thomas Boudier,
Aix-Marseille Université, France
Meghan Driscoll,
University of Texas Southwestern Medical Center, United States

***Correspondence:**
Virginie Uhlmann
uhlmann@ebi.ac.uk

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *15 September 2021*
Accepted: *10 January 2022*
Published: *15 February 2022*

Citation:
Klatzow J, Dalmasso G, Martínez-Abadías N, Sharpe J and Uhlmann V (2022) μMatch: 3D Shape Correspondence for Biological Image Data. Front. Comput. Sci. 4:777615. doi: 10.3389/fcomp.2022.777615

1. INTRODUCTION

Recent progress in microscopy technologies and computational imaging enable the acquisition of large volumes of high-resolution 3D bioimage datasets (Ramirez et al., 2019; Voigt et al., 2019). This increase in imaging quality and throughput makes it possible to visually investigate the tri-dimensional morphology of biological systems at the mesoscopic (Hahn et al., 2020) and microscopic scales (Belay et al., 2021). As a consequence, a growing number of studies focus on quantitatively describing shape variability in the 3D structure of biological objects as observed in biological images (Kalinin et al., 2018; Driscoll et al., 2019; Heinrich et al., 2020).

The quantitative comparison of the shape of objects usually requires that a point-to-point mapping between them, also referred to as correspondence, matching, or registration, is established. This fundamental task is however in general non-trivial because the number of possible mappings scales factorially with the size of the object's surface. The problem of automating shape correspondence has therefore been extensively studied in computer vision (Van Kaick et al., 2011), with applications to medical image analysis (Bône et al., 2018) and evolutionary biology (Martínez-Abadías et al., 2012). Existing 3D shape correspondence methods are however

overwhelmingly designed for macroscopic, highly-stereotypical data such as human organs or skeletal scans and do not translate easily to soft-tissue objects with less predictable morphological variations, extracted from noisy biological images at the micro- or mesoscopic scale. Biological entities such as cells, tissues, and small organisms indeed exhibit a significant degree of variation in morphology, even within groups of similar objects (e.g., cell types, organoids, early embryos). In addition, biological images also generally offer a much lower signal-to-noise ratio than 3D medical acquisition devices or object scanners. Object surfaces extracted from bioimages are thus likely to require extensive "cleaning" prior to shape analysis. Finally, biological experiments most often involve large populations of objects, calling for computationally light and scalable approaches. While several microscopy-specific methods have been proposed for intensity (volume)-based registration (Preibisch et al., 2010; Paul-Gilloteaux et al., 2017), fewer focus on the problem of surface-based registration and existing ones require either manual intervention (Boehm et al., 2011) or specific equipment (Horstmann et al., 2018), or rely on the construction of a shape atlas (Grocott et al., 2016; Toussaint et al., 2021), limiting their use to collections with small amount of shape variability.

In this work, we present μMatch (pronounced *microMatch*), a 3D shape correspondence pipeline tailored to the particularities of structures as they appear in bioimage data. μMatch automates the computation of dense correspondence maps between pairs of 3D mesh surfaces, without the need for any (pseudo-) landmarks. Mesh surfaces are particularly convenient representations for computational geometry and can easily be extracted from segmented voxel data. This input format is therefore ideally suited to the various operations needed for correspondence retrieval and general enough to accommodate a vast range of object geometries, making μMatch amenable to a wide range of bioimage analysis applications. We combine state-of-the-art methods initially developed for graphics and computer vision applications, and package them into a user-friendly, open-source end-to-end Python pipeline. In particular, μMatch contains tools to facilitate each steps required to align collections of 3D shapes of biological objects described as triangular meshes, ranging from mesh cleaning to symmetry identification, correspondence map extraction, and ultimately basic shape analysis. Our pipeline is designed to be computationally light and is thus amenable to the analysis of large shape collections.

The paper is organized as follows: in Section 2, we review the shape correspondence literature that is most relevant to our work and define the notations used through the manuscript. We present μMatch in Section 3 and provide technical details describing each step in the pipeline. We then quantitatively assess in Section 4 the performance of μMatch on a biologically-relevant benchmark dataset for shape correspondence, and demonstrate μMatch's ability to recover previously reported morphological differences in embryonic limb development of wild-type and Apert syndrome mouse models. Finally, we conclude with a discussion in Section 5.

2. THE SHAPE CORRESPONDENCE PROBLEM

2.1. Literature Overview

Biological morphometry historically relies on manually selected homologous landmarks, defined as anatomically unambiguous and consistent features of the object of interest (Bookstein, 1997). Relying on these landmark points provides an implicit form of sparse correspondence between objects, which then allows aligning collections of specimen via Procrustes analysis, a classical strategy consisting of removing geometrical transformations that do not affect the shape of an object (specifically, translations, scaling and rotations) so as to statistically study the extent and nature of shape differences (Rohlf and Slice, 1990).

Landmark-based morphometry, also called geometric morphometrics (Dryden and Mardia, 2016), is actively used at the macroscopic scale in the context of medical imaging (Yeh et al., 2021), anatomy (Finka et al., 2019), taxonomy (Karanovic and Bláha, 2019), and plant science (Lucas et al., 2013). However, when considered at the smaller mesoscopic or microscopic scales, biological objects such as soft tissues and cells in isolation rarely possess well-localized and unambiguously-identifiable features that could reliably act as landmarks, despite having non-trivial shapes. In geometric morphometrics, the fact that landmarks are homologous and that they correspond to the same biological structure or function across different individuals is relevant (Klingenberg, 2008). For the non-rigid, featureless objects most often encountered in microscopy, a more appropriate alternative consists of identifying a dense (continuous) correspondence that only uses geometry and does not rely on the existence of a finite set of unique, localized features derived from functional or evolutionary factors. This type of approach is successfully used in several frameworks in 2D (Laga et al., 2014; Phillip et al., 2021). Extending these methods to 3D shapes is unfortunately challenging when at all possible, and existing solutions impose strong constraints on the topology of input objects and on the way they are parameterized (Srivastava et al., 2010; Koehl and Hass, 2015). As a result, most 3D shape correspondence workflow involving objects extracted from bioimages still rely on manually annotated (pseudo-) landmarks (Boehm et al., 2011; White et al., 2019).

Several algorithms that automatically retrieve a dense correspondence between 3D surfaces have been proposed by the computational geometry and computer graphics communities. Herein it is generally assumed that the shapes to correspond are near-isometric, meaning that they do not exhibit significant deformations. This assumption alone unfortunately makes the vast majority of solutions unsuitable to problems involving collections of biological objects, where natural individual-to-individual variability may fully deviate from isometry. Most popular approaches retrieve a correspondence map that minimizes a measure of distortion, such as the degree of stretching or bending, using continuous optimisation techniques (Schreiner et al., 2004; Sorkine and Alexa, 2007; Schmidt et al., 2019). These methods however suffer from highly non-convex energy landscapes composed of many sub-optimal

local minima, and thus require a relatively good initialization to converge to a good solution. This problem is circumvented by Windheuser et al. (2011), in which a global solution is found using a sophisticated technique to reduce the solution space and then solve the optimization problem using a linear programming approach. The latter approach is guaranteed to yield the lowest distortion mapping and is therefore less likely to suffer from symmetry problems and possible mis-assignments provided that an appropriate distortion function is used. Each matching however then requires hours of processing time even with GPU speedup, making it poorly scalable to large bioimage datasets.

Another class of methods approaches the problem by mapping each of the objects to correspond into intermediate domains and then recovers a mapping between these domains. One such example exploiting the relatively easier task of finding mapping between intermediate domains is provided by Lipman and Funkhouser (2009), available at https://github.com/pedrofreire/shape-matching. There, a conformal parameterisation (i.e., an angle-preserving mapping of the objects to the plane $\mathbb{R}^2$) is used to flip the problem into searching for conformal automorphisms of the plane using Möbius transformations. This method, however, requires good landmarks for initialization and is restricted to shapes of specific topology. Finally, Ovsjanikov et al. (2012) take a more abstract approach by using the scalar function space associated with each object as intermediate domain, in a strategy referred to as functional mapping. The function space is defined as the set of all functions from the object's surface, represented as a mesh $\mathcal{M}$, to the real numbers, $\mathcal{F}_{\mathcal{M}} = \{f : \mathcal{M} \to \mathbb{R}\}$. This last technique is particularly interesting for biological shape correspondence for a number of reasons: firstly, it reduces 3D correspondence to a linear problem that does not require initialization; secondly, it is relatively robust to non-isometry and flexible compared to other continuous approaches as it imposes few constraints on the input shapes; and finally, it provides a computationally efficient means of determining correspondences. For these reasons, we chose to make use of this approach in the μMatch pipeline.

The above literature overview, and the design of the μMatch pipeline, focuses solely on surface-based alignment. An alternative could have been to review and rely on volume-based alignment methods. Our rationale for choosing a surface-based strategy instead is as follows. Volume-based alignment approaches, as implemented for instance in the popular elastix software (Klein et al., 2009) and commonly used in medical imaging, are unable to handle objects with significantly different orientations. Images must then be pre-aligned or a close-enough initialization must be provided (Miao et al., 2016; Yang et al., 2017). While medical image data often exhibit small enough variations in object orientation (due to the acquisition protocols) and small enough sample variation (due to the nature of the objects being imaged) for this issue to be addressed with *ad-hoc* methods, the extent of sample variation and orientation difference in biological experiments cannot be known a priori. In contrast, the performance of surface-based approaches is not affected by the orientation difference and, as a consequence, they do not require any pre-alignment step. A further issue with biological data lies in the anisotropic nature of microscopy image volumes, making additional steps of interpolation mandatory when relying on voxels to align and further complicating the task of volume-based strategies in the case of large orientation differences. Surface-based methods, as they rely on meshes, have the advantage of being blind to the anisotropy of the input data. In addition to these technical aspects, most alignment algorithms for bioimage analysis originate from medical imaging in general and neuroimaging in particular, in which surface-based alignment is now accepted to be superior to volume-based alignment. Surface-based matching has indeed been shown to map borders more accurately between brains than volume-based registration (Brodoehl et al., 2020), further strengthening the case for a surface-based approach. Finally, since biological images most often capture objects that purely have a surface signal (e.g., from membrane stains) and rarely have a conserved inner structure akin to human organs, the information available for alignment is overwhelmingly held in the object's surface, making it all the more relevant to rely on a surface-based method.

2.2. Notations and Problem Formulation

Through the article, we focus on the problem of establishing correspondence between 3D biological objects represented as meshes. We in particular do not study the downstream problems of object segmentation and voxel set meshing. Obtaining object segmentation from possibly noisy biological data can be challenging but is beyond the scope of this paper. This problem has been extensively studied and several robust state-of-the-art methods relying on deep learning, such as the popular 3D U-net (Isensee et al., 2021), are now freely available in user-friendly softwares (Lucas et al., 2021). Once segmented, voxel sets can reliably be meshed using classical methods such as the popular marching cubes algorithm (Lorensen and Cline, 1987), provided in widely-used 3D image processing Python libraries such as scikit-image (van der Walt et al., 2014).

Triangular meshes are discrete approximations of 3D surfaces commonly used as representations for computer graphics and computational geometry. A mesh $\mathcal{M}$ is composed of vertices, edges, and (triangular) faces to form a continuous but not smooth surface, due to the presence of sharp edges connecting any two neighboring faces. A mesh is generally defined by two arrays of numbers, the first one containing the spatial positions of the vertices, $v \in \mathbb{R}^{n_v \times 3}$, and the second one the mesh faces, $f \in \mathbb{N}^{n_f \times 3}$. Increasing the number of vertices of a mesh provides increased resolution and smoothness, at the expense of increased complexity and memory requirements.

Numerous classical Euclidean operators can be extended to manifolds, and as a consequence to discrete meshes, including the gradient ∇ and the Laplacian $\Delta = \nabla^2 = \nabla \cdot \nabla$ operators. The Laplacian is interesting in particular because its eigenfunctions (i.e., the functions that are only scaled by the action of the operator) provide a geometrically informative basis for the scalar space of the manifold. The Laplace-Beltrami operator is the generalization of the Laplacian to triangular meshes. For a mesh of n_v vertices, the Laplace-Beltrami operator is represented as a sparse $n_v \times n_v$ matrix. Because a mesh is not differentiable, such a discretisation requires care in order to reproduce the expected behavior of the Laplacian: the most common formalism for doing

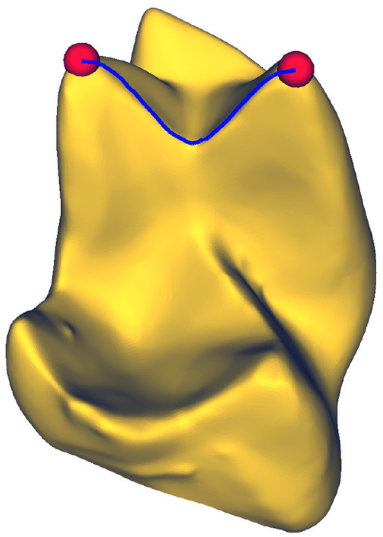


FIGURE 1 | Geodesic path between two points on a surface. A geodesic corresponds to the shortest path connecting any two arbitrary points on a mesh. Molar tooth mesh data from Boyer et al. (2011).

is the cotangent Laplace-Beltrami operator (Pinkall and Polthier, 1993), given by

$$\Delta_{ij} = \begin{cases} \cot(\alpha_{ij}) + \cot(\alpha_{ji}) & j \in \mathcal{N}(i) \\ -\sum_k \Delta_{ik} & i = j \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where α_{ij} is the angle adjacent to the edge ij, $\mathcal{N}(i)$ the connected neighbors of the i^{th} vertex, and cot the cotangent operator.

The shortest distance between any two points on a mesh is given by a geodesic path, as illustrated in **Figure 1**. Formally, given a triangular mesh $\mathcal{M}$ with associated metric g, the geodesic path x parameterized by $t \in \mathbb{R}$ between two points a and b on the mesh is defined as

$$x(t) = \underset{x}{\operatorname{argmin}} \int_a^b \sum_{\mu\nu} g_{\mu\nu} \frac{dx_\mu}{dt} \frac{dx_\nu}{dt} dt, \tag{2}$$

where $\mu$and ν correspond to coordinates on the mesh.

Two meshes are equivalent if there is a map between them that preserves the metrics, and therefore the geodesic distances, on them. The degree of deformation induced by a map can thus be measured by calculating the extent to which geodesic distances are altered by the map, captured in the geodesic matrix $\mathbf{G}_{\mathcal{M}}$. The geodesic matrix is constructed by computing all pairwise geodesic distances such that $[\mathbf{G}_{\mathcal{M}}]_{ij}$ contains the geodesic distance between the two vertices i and j on the mesh $\mathcal{M}$.

A symmetry of a surface is defined as a self-mapping (i.e., automorphism) Ψ of the mesh that leaves the geodesic matrix unchanged, which is formally expressed as $\Psi : \mathcal{M} \rightarrow \mathcal{M}$ such that $[\mathbf{G}_{\mathcal{M}}]_{x,y} = [\mathbf{G}_{\mathcal{M}}]_{\Psi(x),\Psi(y)}$ for every $x, y \in \mathcal{M}$.

Finally, a correspondence Φ between two meshes is formally defined as a mapping that assigns each vertex on a first mesh $\mathcal{M}_1$ to each vertex on a second mesh $\mathcal{M}_2$ as $\Phi : \mathcal{M}_1 \rightarrow \mathcal{M}_2$. A functional mapping $T : \mathcal{F}_{\mathcal{M}_1} \rightarrow \mathcal{F}_{\mathcal{M}_2}$ maps the space of scalar functions on one mesh ($\mathcal{F}_{\mathcal{M}_1}$) to the scalar functional space on the other ($\mathcal{F}_{\mathcal{M}_2}$). As will be extensively discussed through the paper, the functional mapping provides a useful way of representing the correspondence between two meshes. In order to use it, we need a concrete representation in the form of a finite $(k \times k)$ correspondence matrix, denoted as $\mathbf{C}_\Phi$.

3. μMATCH PIPELINE

We hereafter detail each step of the μMatch pipeline that automatically retrieves a one-to-one mapping between pairs of biological objects which surfaces are represented as triangular meshes. The considered objects can be of any nature as long as a mapping reasonably exists between them. To provide an intuitive, non-biological toy example of what a reasonable mapping means: while any four-legged animal such as cats and dogs can plausibly be matched onto one another, they cannot not be plausibly matched with a snake. It is worth noting that there is no formal criterion to determine whether two shapes can meaningfully be put in correspondence, nor for evaluating the quality of the resulting correspondence in the absence of a ground truth. The quality of a matching is therefore usually left to qualitative evaluation.

The μMatch pipeline, summarized in **Figure 2**, is composed of three main modules. It starts with a pre-processing stage in which the meshes are prepared and cleaned and some important quantities are pre-computed. The second step is the matching algorithm itself, which exploits the functional mapping strategy. Finally, the retrieved correspondence map can be further used for shape analysis. μMatch is implemented in Python 3.7 and is available at *github.com/uhlmanngroup/muMatch*. In addition to the code itself, we also provide sample data and a script exemplifying the use of the pipeline. For ease of use, all tunable parameters involved in μMatch are gathered in a `.yml` file along with a description of their meaning and range.

3.1. Pre-processing

To make them amenable to processing with μMatch, the input objects must satisfy a number of technical requirements. First, each object's surface must be represented in the form of a triangular mesh. These meshes must additionally be both connected and manifold. Being connected implies that, given any two vertices in the mesh, it is possible to find a path consisting of a subset of the mesh edges (i.e., the sides of the faces) that links the two. The manifold condition imposes that the mesh must represent a physically realizable continuous surface. More specifically, it means that each edge in the mesh should be incident to at most two faces, and that the faces attached to a vertex should form an open disk or half-disk around the vertex. An example of non-manifold mesh may involve two parts that are connected by a single vertex, or the presence of an internal face. These different notions are illustrated in **Figure 3**.

In addition, the matching process assumes that the correspondence map is a bijection, meaning that it defines

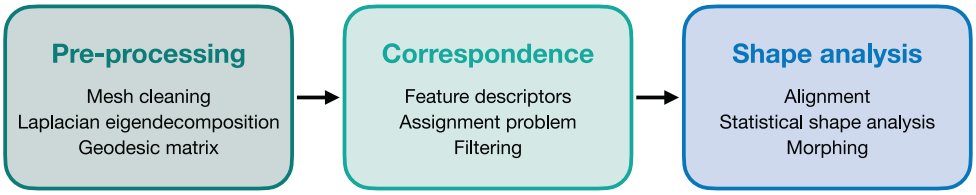


FIGURE 2 | Overview of μMatch. The pipeline, composed of three main steps, automatically retrieves a dense correspondence map between two arbitrary surfaces of biological objects represented as triangular meshes. Individual substeps are listed in each of the boxes.

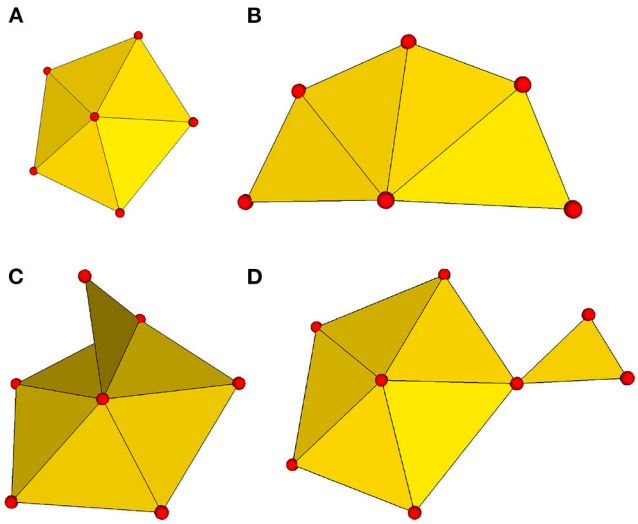


FIGURE 3 | Connectivity and manifold properties. Connectivity attributes expected in a manifold mesh: **(A)** internal vertex with a full loop of surrounding faces; **(B)** boundary vertex with a partial loop of surrounding faces. Two examples of non-manifold connectivity: **(C)** sub-meshes connected by only a single vertex; **(D)** a single edge shared by more than two triangular faces.

a way to match $\mathcal{M}_1$ to $\mathcal{M}_2$ as much as a way to match $\mathcal{M}_2$ to $\mathcal{M}_1$. It is therefore crucial that the pairs of meshes have the same coverage, which implies that for each point in $\mathcal{M}_1$, a corresponding point exists in $\mathcal{M}_2$ and vice versa. Finally, it is important for all meshes in the collection to be constructed by sampling as uniformly as possible the object's surface. This translates to vertices in the mesh being equally spaced on each surface, or equivalently to triangular faces having a constant area. μMatch makes use of a number of geometric quantities associated with each mesh, including the geodesic matrix and the Laplace-Beltrami eigen-decomposition introduced in Section 2.2, that may be adversely affected by a non-uniform sampling.

In μMatch, input meshes are cleaned using PyMeshFix 0.14.1 (Attene, 2010), which offers built-in functionalities to remove common errors in triangular meshes, including degenerate and intersecting faces. As a result, output meshes are manifold and watertight with a single connected component. Meshes are subsequently resampled to a user-defined number of vertices N that is smaller or equal than the total number of vertices in the smallest of the input meshes. Importantly, this resampling step processes all meshes to have the same number of vertices regardless of the original number of vertices they were composed of, relieving users from having to handle this constraint. The value of N should be chosen so as to aim at having fine enough meshes to capture important shape features, while limiting the number of vertices to what is necessary in order to avoid overburdening the matching process later.

A final preprocessing step in μMatch, consists of calculating the geodesic matrix of each input mesh using Lib-igl (Jacobson and Panozzo, 2018). Although Lib-igl implements the fast exact geodesic algorithm (Mitchell et al., 1985), it is still computationally demanding with a run-time of $O(N^2 \log N)$ and memory requirement of $O(N^2)$, with N the number of vertices in the mesh. For the sake of computational efficiency, we therefore provide a custom data preparation script that precomputes the geodesic matrices for the whole collection of objects and save them to disk prior to the correspondence pipeline.

3.2. Correspondence

The core of μMatch is a landmark-free dense correspondence algorithm adapted from Ovsjanikov et al. (2012), Litany et al. (2017), and Halimi et al. (2019). The correspondence matrix, describing the final mapping, is built from a collection of feature descriptors and refined through filtering steps. The overall workflow is summarized in **Figure 4**.

3.2.1. Feature Descriptors

Surface matching begins with the calculation of point-wise feature descriptors, referred to as signature functions. Signature functions describe the intrinsic geometry of an object and will therefore return similar values at geometrically similar points on each mesh to be matched. A number of such descriptors have been developed and proposed for the specific purpose of shape correspondence. The first and perhaps most intuitive one is the Gaussian curvature of the mesh, defined as the product of the two principal curvatures, calculated at different scales. More sophisticated techniques include the heat kernel signature (HKS) (Sun et al., 2009), based on solutions of the classical heat equation $\partial_t T = \Delta T$, where Δ is the Laplace-Beltrami operator and T a temperature. The rough idea behind the HKS is to express the remaining temperature after some time t for an initial "heat impulse" fully concentrated at a given point of the mesh. The HKS can be computed for multiple values of t at each point on a mesh to generate a collection of feature descriptors. In practice,

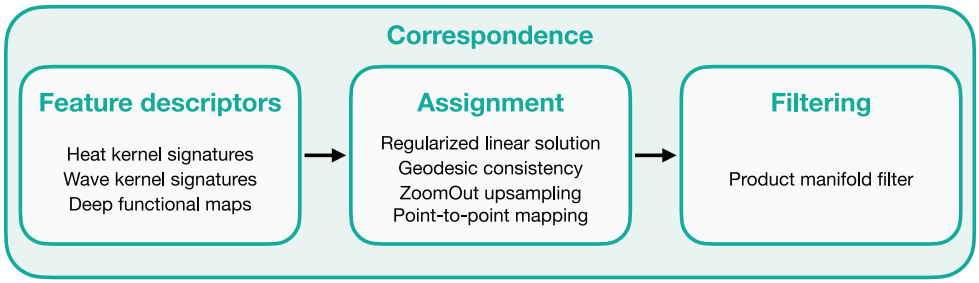


FIGURE 4 | Overview of μMatch's correspondence workflow. The main elements involved in each steps are listed in the corresponding boxes.

the HKS can be calculated efficiently as

$$h_t(x) = \sum_{n=0}^{N} e^{-\lambda_n t}\phi_n(x)^2, \tag{3}$$

where $\{\lambda_i, \phi_i\}$ are the pairs of eigenvalues and eigenvectors of the Laplace-Beltrami operator (calculated with Lib-igl in μMatch).

The wave kernel signature (WKS) (Aubry et al., 2011) is closely related to the HKS but instead uses the Schrödinger wave equation $(i\partial_t + \Delta)\psi = 0$, which ordinarily governs the dynamics of particles in quantum mechanics. Unlike HKS, WKS assume an approximate particle energy and then determines the long time (i.e., $t \to \infty$) averaged probability distribution for finding the particle in a particular location on the surface. Sampling these distributions for different values of particle energy once again produces a collection of feature descriptors at each point on the mesh. For $f_\epsilon(E) = \exp\left(-0.5(\epsilon - \log(E))^2/\sigma^2\right)$, where ϵ is the particle energy and σ a scale factor set by the difference between the eigenvalues, it can be shown that the WKS is obtained as

$$s_\epsilon(x) = \sum_{n=0}^{N} f_\epsilon(\lambda_n)^2\phi_n(x)^2. \tag{4}$$

Examples of several HKS and WKS, exhibiting these descriptors' ability to capture geometrically similar features across comparable objects, are illustrated in **Figure 5**.

In order to obtain robust and quickly-computed feature descriptors for use on biological objects exhibiting potentially subtle shape features, μMatch combines HKS and WKS. In practice these quantities are calculated using only the eigenvalues and eigenvectors of the Laplace-Beltrami operator as input, together with the number of functions desired. The total number of extracted HKS and WKS signature functions must be greater or equal as the dimension of the functional space, which corresponds to the size of the correspondence matrix to extract. This parameter can be freely adjusted in μMatch and is set to 100 by default.

A final step in the computation of signature functions consists of passing them through a neural network to improve their overall quality in a process called deep functional maps. Following Halimi et al. (2019), μMatch implements a 7-layer ResNet, which is trained in an unsupervised fashion. Each layer acts on the signature functions by first taking weighted linear combinations of the different signature functions and then passing the result through a non-linear activation function (ReLU). The weights in the linear combination are adjustable parameters that are tuned during the training process such that the output signature functions span the functional space better and therefore produce correspondence maps with lower distortion. Note that while the loss is formulated on the geodesic distance, the neural network is however constrained by the input signature functions. Minimizing the geodesic distance between the two inputs therefore amounts to minimizing distortion subject to aligning the signature functions. As such, the optimization is carried out on intrinsic shape properties (as captured by the signature functions) and not on the geodesic matrix itself. In order to train the network, a subset of the collection of objects to be put in correspondence must serve as training set. As ever, the size of the training set is a trade-off between the ability of the trained network to effectively generalize and the training time, and there is no universal rule to determine how many objects should be included for training. In practice, however, all objects may be used whenever the collection is small. For collections composed of more than a hundred of meshes, a representative subset can be chosen as the training set to speed up computations. We recall that preparing the training set does not require any manual annotation since the network is trained in an unsupervised manner. For each mesh in the training set, the geodesic matrix, the Laplace-Beltrami eigenvectors, an array area encoding the area around each vertex, and the feature descriptors of each mesh are precomputed and saved to a single TensorFlow `.tfrecords` file for fast data access.

3.2.2. Assignment Problem

The correspondence matching algorithm used in μMatch is based on a method known as functional mapping (Ovsjanikov et al., 2012, 2016) and entirely relies on the computed feature descriptors. The core principle of functional maps is that any mapping between two surfaces induces a corresponding linear mapping between their functional spaces. Formally, this can be demonstrated as follows. Assuming that two surfaces $\mathcal{M}_1$ and $\mathcal{M}_2$ are put in correspondence by a smooth mapping $\Phi : \mathcal{M}_1 \to \mathcal{M}_2$, one can define a space of scalar functions on the surface $\mathcal{M}_\nu$ as $\mathcal{F}_{\mathcal{M}_\nu} = \{f : \mathcal{M}_\nu \to \mathbb{R}\}$ for $\nu = 1, 2$. These spaces can be shown to be infinite dimensional linear vector spaces and the

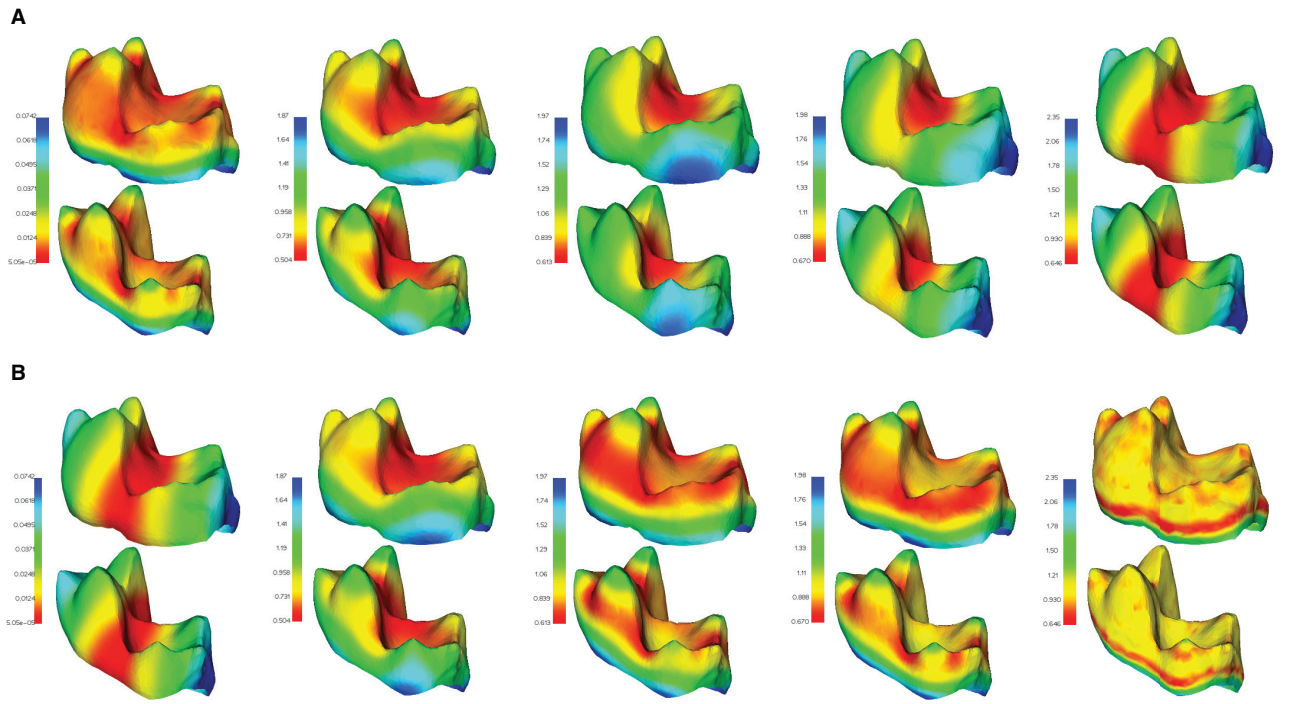


FIGURE 5 | Feature descriptors computed in μMatch. A subset of the descriptors computed from the eigen-decomposition of the Laplacian operator are shown in two different meshes of the TEETH dataset: **(A)** heat kernel signatures (HKS) for different values of t; **(B)** wave kernel signatures (WKS) for different values of ϵ. Both types of descriptors can be observed to behave similarly in corresponding areas of the two meshes. Molar teeth mesh data from Boyer et al. (2011).

smooth mapping Φ thus induces a linear map $T_\Phi : \mathcal{F}_{\mathcal{M}_1} \rightarrow \mathcal{F}_{\mathcal{M}_2}$ through $T_\Phi(f) = f \circ \Phi^{-1}$, called functional mapping. The linear map T_Φ can be represented as a finite correspondence matrix $\mathbf{C}_\Phi$ by choosing a (finite) set of k basis functions on the vector spaces $\mathcal{F}_{\mathcal{M}_1}$ and $\mathcal{F}_{\mathcal{M}_2}$. The eigenfunctions of the Laplace-Beltrami operator, $\{\phi_i\}_{i=1}^k$, can be shown to form a complete orthogonal basis of each respective functional space (Ovsjanikov et al., 2012) and encode spatial resolution (also referred to as frequency) when ordered according to their eigenvalues, denoted $\{\lambda_i\}_{i=1}^k$, making their choice geometrically meaningful. Because only the k first eigenfunctions are selected, they are referred to as the reduced spectral basis.

A correspondence should first and foremost preserve the signature functions. Given a set of n feature descriptors for each mesh, $\mathbf{A} \in \mathbb{R}^{k \times n}$ and $\mathbf{B} \in \mathbb{R}^{k \times n}$ expressed in the reduced spectral basis, this requirement implies that $\mathbf{C}_\Phi \mathbf{A} \simeq \mathbf{B}$. In the case of perfect isometry, the functional mapping can be shown to commute with the Laplace-Beltrami operator such that $\mathbf{C}_\Phi \mathbf{L}_1 = \mathbf{L}_2 \mathbf{C}_\Phi$, with $\mathbf{L}_\nu$ the Laplace-Beltrami operator expressed in the spectral basis of $\mathcal{M}_\nu$, simply corresponding to a diagonal matrix of the eigenvalues. Whilst exact equality is no longer true in the general case, this relation still approximately holds and can be exploited as a regulariser. The correspondence matrix can be retrieved by solving the regularized linear least squares problem

$$\mathbf{C}_\Phi = \arg\min_{\mathbf{C}} \|\mathbf{C}\mathbf{A} - \mathbf{B}\|^2 + \|\delta\mathbf{L} \odot \mathbf{C}\|^2, \tag{5}$$

where $\odot$ is the Hadamard product and $[\delta\mathbf{L}]_{ij} = \lambda_i^{\mathcal{M}_2} - \lambda_j^{\mathcal{M}_1}$, with $\left\{\lambda_\mu^{\mathcal{M}_\nu}\right\}_{\mu=1}^k$ the eigenvalues of the Laplace-Beltrami operator on $\mathcal{M}_\nu$.

In the presence of any intrinsic symmetries in the mesh, a subset of eigenfunctions called the anti-symmetric space are undistinguishable up to a flip of sign. The signature functions therefore do not provide sufficient information to unambiguously determine how this subset of the function space is mapped between the two meshes. This can lead to parts of the object being mapped with different orientations, leaving large tear discontinuities in-between. To solve this, μMatch implements an additional step after solving (Equation 5) in order to ensure that the resulting map has low distortion. We begin by noting that, much like any scalar function can be decomposed using the spectral basis, so too can any bivariate function, $F : \mathcal{M}_\nu \times \mathcal{M}_\nu \mapsto \mathbb{R}$, the result being a $k \times k$ array. Also, given such a bivariate function F on $\mathcal{M}_1$ and a mapping $\Phi : \mathcal{M}_1 \mapsto \mathcal{M}_2$, Φ maps F to a bivariate function F_Φ on $\mathcal{M}_2$, defined by $F_\Phi(\cdot,\cdot) = F(\Phi^{-1}(\cdot), \Phi^{-1}(\cdot))$. When expressed in the spectral basis with the correspondence matrix, it reduces to the matrix product $\mathbf{F}_\Phi = \mathbf{C}_\Phi \mathbf{F} \mathbf{C}_\Phi^T$. The class of bivariate functions that is particularly interesting for the purpose of shape correspondence is the geodesic matrix and functions derived from it. In μMatch, we don't consider the geodesic matrix directly, but rather its Gaussian at several scales σ, defined as

$\mathbf{K}_\sigma^{\mathcal{M}_v} = \exp(-\mathbf{G}^2_{\mathcal{M}_v}/2\sigma^2)$, which better encodes neighborhood information. Thus, we expect

$$\mathbf{C}_\Phi \mathbf{K}_\sigma^{\mathcal{M}_1} \mathbf{C}_\Phi^T \simeq \mathbf{K}_\sigma^{\mathcal{M}_2}. \tag{6}$$

The $\mathbf{K}_\sigma^{\mathcal{M}_v}$ matrices contain off-diagonal entries that provide information on how different eigenfunctions should relate to one-another, including for the anti-symmetric ones that are not captured by the signature functions. In μMatch, a few fixed values of σ are chosen corresponding to $\sigma \in [0.25\mu(\mathbf{G}_\mathcal{M}),\ 0.5\mu(\mathbf{G}_\mathcal{M}),\ 0.75\mu(\mathbf{G}_\mathcal{M})]$, where $\mu(\mathbf{G}_\mathcal{M})$ is the mean surface distance defined as

$$\mu(\mathbf{G}_\mathcal{M}) = \frac{1}{A^2_\mathcal{M}} \int_\mathcal{M} \mathbf{G}_\mathcal{M}(x,y)\, d^2x\, d^2y, \tag{7}$$

for a mesh $\mathcal{M}$ with geodesic matrix $\mathbf{G}_\mathcal{M}$ and surface area $A_\mathcal{M}$.

Decreasing the dimension of the spectral basis, captured by the number of basis functions k, increases the stability of the solution of Equation (5). It is therefore better to solve for the correspondence matrix $\mathbf{C}_\Phi$ at relatively low dimension, generally lying in the range of $k = 5$ to 12 (μMatch uses $k = 8$ by default), and then scale up in a process known as ZoomOut Upsampling (Melzi et al., 2019). The downside of working at low dimensions is that much of the off-diagonal information is lost by the omission of higher frequency components from the matrices $\mathbf{K}_\sigma^{\mathcal{M}_v}$. Therefore, in order to retain the high frequency information that is required for computing the final $k_{\max}$-dimensional correspondence matrix, we first compute $\mathbf{K}_\sigma^{\mathcal{M}_v}$ matrices at the full dimension $k_{\max}$, and from this derive the following three reduced $(k \times k)$ matrices

$$[\mathbf{J}^{(0)}_{\mathcal{M}_v,\sigma}]_{ij} = [\mathbf{K}_\sigma^{\mathcal{M}_v}]_{ij}, \tag{8}$$

$$[\mathbf{J}^{(1)}_{\mathcal{M}_v,\sigma}]_{ij} = \textstyle\sum_{m=k+1}^{k_{\max}} [\mathbf{K}_\sigma^{\mathcal{M}_v}]_{im} [\mathbf{K}_\sigma^{\mathcal{M}_v}]_{mj}, \tag{9}$$

$$[\mathbf{J}^{(2)}_{\mathcal{M}_v,\sigma}]_{ij} = \textstyle\sum_{m,n=k+1}^{k_{\max}} [\mathbf{K}_\sigma^{\mathcal{M}_v}]_{im} [\mathbf{K}_\sigma^{\mathcal{M}_v}]_{mn} [\mathbf{K}_\sigma^{\mathcal{M}_v}]_{nj} \tag{10}$$

for $i,j \in \{1,\ldots,k\}$. The reduced matrices $\mathbf{J}^{(1)}_{\mathcal{M}_v,\sigma}$ and $\mathbf{J}^{(2)}_{\mathcal{M}_v,\sigma}$ capture off-diagonal information related to the first k eigenfunctions that would otherwise be lost in the $k \times k$ version of $\mathbf{K}_\sigma^{\mathcal{M}_v}$ (corresponding to $\mathbf{J}^{(0)}_{\mathcal{M}_v,\sigma}$). A $k \times k$ correspondence matrix $\mathbf{C}_\Phi^{(k)}$ can then be obtained by solving the continuous optimisation problem

$$\mathbf{C}_\Phi^{(k)} = \arg\min_{\mathbf{C}} \sum_\sigma \sum_{i=0}^{2} \left\| \mathbf{C}\mathbf{J}^{(i)}_{\mathcal{M}_1,\sigma} \mathbf{C}^T - \mathbf{J}^{(i)}_{\mathcal{M}_2,\sigma} \right\|_F^2, \tag{11}$$

using the solution of Equation (5) as initial value.

The final, full resolution correspondence matrix is obtained by iteratively increasing k as follows. First, given $\mathbf{C}_\Phi^{(k)}$, we solve for the point-to-point map by computing a transition matrix $\mathbf{Q}$ given as

$$\mathbf{Q} = \mathbf{A}_{\mathcal{M}_1} \phi^{(k)}_{\mathcal{M}_1} (\mathbf{C}_\Phi^{(k)})^T (\phi^{(k)}_{\mathcal{M}_2})^T, \tag{12}$$

and a probability matrix $\mathbf{P}$ given as

$$[\mathbf{P}]_{ij} = [\mathbf{Q}^2]_{ij} / \sum_k [\mathbf{Q}^2]_{ik}. \tag{13}$$

The point-to-point mapping Φ is then retrieved by passing P to a linear assignment algorithm (Crouse, 2016), which assigns the indices of vertices of one mesh to those of the other, yielding the maximum probability point-wise correspondence. Conversely, given the point-to-point mapping $\Phi : \mathcal{M}_1 \mapsto \mathcal{M}_2$, the associated correspondence matrix of dimension $k+1$ can be computed as

$$\mathbf{C}_\Phi^{(k+1)} = (\phi^{(k+1)}_{\mathcal{M}_1})^T \mathbf{A}_{\mathcal{M}_1} \Pi\, \phi^{(k+1)}_{\mathcal{M}_2}, \tag{14}$$

where $\mathbf{A}_{\mathcal{M}_1}$ is the mesh mass matrix encoding the area around each vertex on $\mathcal{M}_1$, and $\phi^{(k)}_{\mathcal{M}_v}$ are matrices which columns correspond to the first k Laplace-Beltrami eigenvectors of the mesh $\mathcal{M}_v$. Finally, Π is the matrix representation of Φ given by

$$[\Pi]_{ij} = \begin{cases} 1 & \Phi(i) = j \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

ZoomOut Upsampling thus proceeds to alternate between solving for Π and $\mathbf{C}_\Phi$, incrementing the dimension by one each time we solve for $\mathbf{C}_\Phi$ and repeating until $k = k_{\max}$, where $k_{\max}$ is the dimension of the reduced spectral basis, chosen such that it provides sufficient resolution for an accurate mapping.

3.2.3. Filtering

The occasional assignment errors and discontinuities that may remain after the correspondence procedure can be filtered out using a technique known as product manifold filter (PMF, Vestner et al., 2017). The PMF aims at producing a bijective and continuous correspondence map by using kernel density estimation. Assuming a noisy and potentially sparse set of I correspondences $\{x_i \to y_i\}_{i=1,\ldots,I}$ and a kernel κ (i.e., window function), the function $F : \mathcal{M}_1 \times \mathcal{M}_2 \to \mathbb{R}$ defined as

$$F(x,y) = \sum_{i=1}^{I} \kappa(d(x,x_i))\ \kappa(d(y,y_i)), \tag{16}$$

where d is a distance measure (the geodesic distance in our case). The expression (Equation 16) will be maximized when the distance between y and y_i is similar to that between x and x_i for each $i = 1,\ldots,I$.

A clean mapping $\Phi : x \to y$ can thus be retrieved when y maximizes (Equation 16), which can be expressed in matrix form as $\mathbf{F}_\sigma = \mathbf{K}_\sigma^{\mathcal{M}_1} (\mathbf{K}_\sigma^{\mathcal{M}_2})^{\mathrm{T}}$, where $\mathbf{K}_\sigma^{\mathcal{M}_v} = \exp(-\mathbf{G}^2_{\mathcal{M}_v}/2\sigma^2)$ and $\sigma = 0.3\mu(\mathbf{G})$ as recommended in Vestner et al. (2017) (see Equation 7). The matrix $\mathbf{F}$ is once again passed to a linear assignment algorithm to yield a mapping of vertex indices from

one mesh to the other, as done for the probability matrix $\mathbf{P}$ in Section 3.2.2, to obtain an updated correspondence.

Ultimately, for a pair of meshes fed into the correspondence pipeline, μMatch returns a correspondence in the form of two matrices, $\mathbf{N}_1$ and $\mathbf{N}_2$, where $\mathbf{N}_1$ are the indices of vertices on the first mesh and $\mathbf{N}_2$ on the second, such that $[\mathbf{N}_1]_n \mapsto [\mathbf{N}_2]_n$ for each $n \in \mathbf{N}_1$. These arrays are saved to disk for reuse in the shape analysis module or outside of μMatch.

3.3. Shape Analysis

3.3.1. Alignment and Statistical Shape Analysis

μMatch allows putting collections of objects in correspondence in order to computationally retrieve the average shape and morphological variations within the collection relying on Procrustes analysis (Kendall, 1989). All objects in the collections are first centered at the origin and scaled to be of unitary root mean square norm. An arbitrary object in the collection is then selected to act as reference, and correspondence maps between this reference and all other objects in the collection are extracted. The final step is to find the best alignment of each object in the collection onto the reference by rotating it. The optimal rotation angle is retrieved by solving the classical orthogonal Procrustes problem (Gower and Dijksterhuis, 2004). Once all objects have been aligned onto the reference, a new reference is obtained by calculating the mean across the entire collection for each corresponding points. The whole process is then repeated, now aligning every shape in the collection to this new reference, and a new updated mean is computed until convergence. The final result is taken as mean shape for the collection.

Relying on the mean of the collection and each object aligned to it, deviations from the mean can be extracted as follows. For each point on the mean shape, μMatch calculates the standard deviation of corresponding points across the entire collection. This results in a scalar function on the reference shape indicating where morphological variations occur and to which extent, and can be visualized as a colormap on the mean shape mesh relying on the vedo visualization library (Musy, 2021). In μMatch, the script implementing this procedure takes as input a directory containing a collection of meshes together with a path to their pre-computed correspondences. The correspondences need only be calculated between the reference shape and each of the other shapes in the collection.

3.3.2. Morphing

Once correspondence between two shapes is established, a continuous morphing between them can be extracted by first aligning the two shapes via translation and rotation (similar to the procedure described in Section 3.3.1, but without scaling), then calculating the geodesic path connecting the two meshes, and sampling shapes along that path. A geodesic being defined as the shortest path connecting any two points in a space, its computation solely requires a notion of distance between points in that space. In **Figure 1**, we illustrated a geodesic path on a mesh $\mathcal{M}$, where samples along the path correspond to positions on the mesh. In the case of morphing, the space of interest is instead that of surfaces represented by meshes of fixed connectivity meeting the conditions introduced in Section 3.1: points in the space correspond to the vertex positions and the distance is determined by their connectivity. Samples along the resulting path therefore correspond to meshes. Continuous morphing, going beyond shape analysis of collections of equivalent samples, is of particular interest for image-based modeling, for instance in the context of developmental studies. In **Figure 6A**, we illustrate as an example the surfaces of a developing mouse limb bud synthetically generated by interpolating between a young and an old limb bud sample. While the older limb bud sample has features (*e.g.*, distributions of surface curvature) for which there is no simple mapping onto the youngest limb bud, the collection of extracted signature functions (**Figures 6B,C**) manages to capture the local arrangement of these geometrical details with respect to the overall bud shape and allows mapping them to the younger, smoother limb bud in a qualitatively sensible way. As a result, the morphing obtained based on this correspondence produce a visually realistic evolution from one shape onto the other.

Technically, the continuous morphing is obtained following the algorithm proposed in Kilian et al. (2007). Consider a mesh $\mathcal{M}_1$ with vertices v, edges E, and a vector field $\mathbf{X}$ describing the deformation of $\mathcal{M}_1$ onto some other mesh $\mathcal{M}_2$. The vector field $\mathbf{X}$ is given by difference vectors between the two meshes and thus assumes that a correspondence between $\mathcal{M}_1$ and $\mathcal{M}_2$ has been calculated. Then, the norm ϵ, defined as

$$\epsilon(\mathbf{X}) = \sum_{i,j \in E} \langle \mathbf{X}_i - \mathbf{X}_j,\ v_i - v_j \rangle^2, \tag{17}$$

measures the deformation between the two meshes. In order to generate a sequence of meshes morphing the meshes onto one another, the quantity (Equation 17) is minimized through a multi-scale continuous optimisation procedure, starting with low mesh resolution and few intermediate steps and sequentially increasing both. The final geodesic path consists of a set of intermediate meshes that reflects the continuous deformation of one surface onto the other.

4. EXPERIMENTS

4.1. Quantitative Validation

In order to quantitatively validate μMatch, we use a teeth scan dataset originally introduced by Boyer et al. (2011) and available at www.wisdom.weizmann.ac.il/~ylipman/CPsurfcomp/ and commonly exploited to assess 3D morphometry algorithms. The dataset includes 45 meshes corresponding to mandibular second molars of a variety of prosimian primates and their close non-primate evolutionary relatives, summarized in **Table 2**. Each specimen is annotated with a set of 16 manually-determined landmarks placed by expert morphometricians that can be used as ground-truth to assess the quality of an automatically-retrieved correspondence. Although this dataset is composed of objects toward the macroscale of the spectrum, it is one of the rare available resources of biological objects for which a ground truth sparse correspondence is available and is as such an excellent candidate for benchmarking. Performance assessment is then carried out following the Princeton benchmark protocol

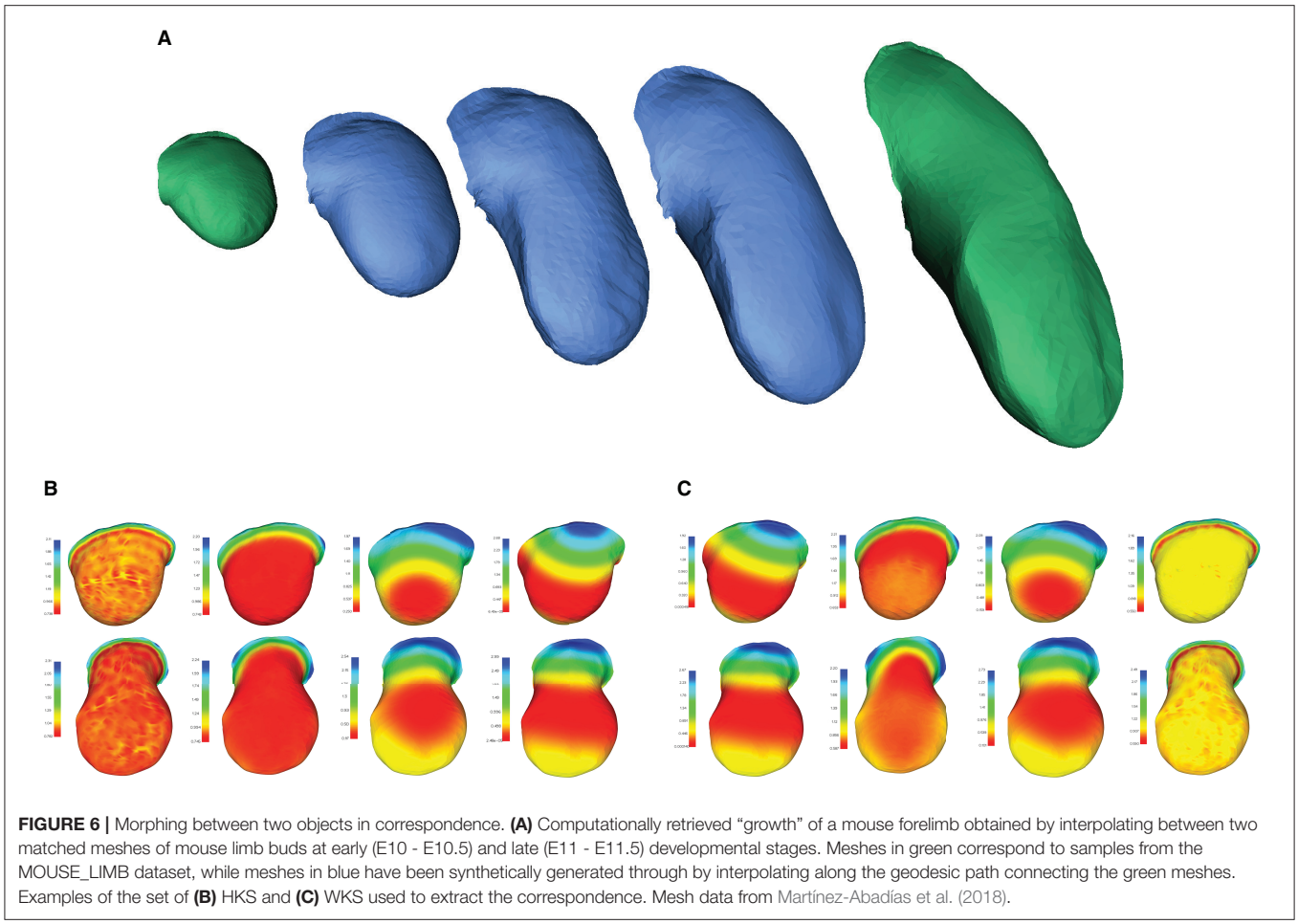


FIGURE 6 | Morphing between two objects in correspondence. **(A)** Computationally retrieved "growth" of a mouse forelimb obtained by interpolating between two matched meshes of mouse limb buds at early (E10 - E10.5) and late (E11 - E11.5) developmental stages. Meshes in green correspond to samples from the MOUSE_LIMB dataset, while meshes in blue have been synthetically generated through by interpolating along the geodesic path connecting the green meshes. Examples of the set of **(B)** HKS and **(C)** WKS used to extract the correspondence. Mesh data from Martínez-Abadías et al. (2018).

(Shilane et al., 2004; Kim et al., 2011) by computing the mean geodesic error (GE) expressed as

$$\mathrm{GE} = \frac{1}{A_{\mathcal{M}_2}^{1/2}} \sum_{i=1}^{16} \mathbf{G}_{\mathcal{M}_2}\left(\Phi(l_i^1), l_i^2\right), \tag{18}$$

where $\{l_i^1\}_{i=1}^{16}$ and $\{l_i^2\}_{i=1}^{16}$ are the ground-truth landmarks on $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, Φ is the point-to-point mapping, and $\mathbf{G}_{\mathcal{M}_2}$ and $A_{\mathcal{M}_2}$ are the geodesic matrix and the surface area of the mesh $\mathcal{M}_2$, respectively. The GE therefore measures how well the main features of the objects have been mapped, regardless of the amount of deformation there may be elsewhere on the surface.

Several processing steps of the μMatch correspondence module (**Figure 4**), namely the deep functional maps and product manifold filter, aim at improving the final correspondence but are not strictly necessary. While we do recommend including them whenever possible, these steps do increase computation time and the extent of the improvement they will bring cannot be predicted in general for any arbitrary dataset. To provide users with an example of the cost and gains involved, we have quantified the quality of the retrieved correspondence in different settings, by enabling and disabling the deep functional maps and product manifold filter in the μMatch pipeline. In **Table 1**, we report the average GE between ground-truth landmarks and correspondence points obtained automatically with μMatch for the whole dataset. We also indicate the average runtime to establish correspondence in any given pair. A breakdown of the average GE by species and the cumulative error curves are provided in **Figure 7A** and **Table 2**, respectively. We observe the respective merits of the deep functional maps and product manifold filter to be as follows. The deep functional maps improve the signature functions but do not impose smoothness in the final correspondence, resulting in large GE variance. While the correspondence runtime is not affected by deep functional maps since the network is trained during preprocessing, a significant extra amount of compute (∼3 h for this dataset) is required ahead of the matching itself, impacting the total duration of the pipeline. In contrast, the product manifold filter ensures that the final correspondence is smooth and does not feature major tears or discontinuities, resulting in small GE variance. It is limited by the correspondence quality that can be obtained from the original signature functions alone and doubles the runtime per pair, but as a trade-off requires no prior computation or training. The combination of the two yields

TABLE 1 | Summary of results on the TEETH dataset.

	No DFM, no PMF	DFM, no PMF	No DFM, PMF	DFM, PMF
Average GE	10.7± 14.8	9.3 ± 12.8	6.5 ± 9.5	**4.8 ± 3.8**
Average runtime (seconds)	**5**	**5**	11	11

The average and standard deviation of the mean geodesic error (GE, Equation 18) between ground truth landmarks and predicted correspondence obtained with μMatch is reported for different setting including or not the deep functional maps (DFM) and product manifold filter (PMF) steps, along with the average runtime per pair (the complete dataset is composed of 990 unique pairs). The GE is reported in units of normalized geodesic length, with lower values indicating better results. The complete μMatch pipeline corresponds to the last column (DMF, PMF). Best results for each metrics are highlighted in bold.

TABLE 2 | Per-specie breakdown of results on the TEETH dataset.

Species	Human	Bonobo	Chimpanzee	Orangutan	Gorilla
# samples	10	4	12	10	9
GE (No DFM, no PMF)	7.4 ± 12.1	5.8 ± 8.3	8.3 ± 12.8	17.0 ± 21.1	7.0 ± 9.5
GE (DFM, no PMF)	7.2 ± 11.9	5.8 ± 8.3	8.2 ± 12.7	12.0 ± 15.7	6.9 ± 9.3
GE (No DFM, PMF)	**3.2 ± 2.1**	3.3 ± 2.1	**3.7 ± 2.6**	11.9 ± 17.6	4.6 ± 2.9
GE (DFM, PMF)	**3.2 ± 2.1**	**3.2 ± 1.9**	**3.7 ± 2.5**	**5.0 ± 3.6**	**4.3 ± 2.8**

For each class, $0.5 \times n \times (n-1)$ unique pairs can be put in correspondence, where n is the number of samples. The average and standard deviation of the mean geodesic error (GE, Equation 18) between ground truth landmarks and predicted correspondence obtained with μMatch is reported for different setting including or not the deep functional maps (DFM) and product manifold filter (PMF) steps. The GE is reported in units of normalized geodesic length, with lower values indicating better results. The complete μMatch pipeline corresponds to the last row (DMF, PMF). Best results for each specie are highlighted in bold.

the best results, as it allows obtaining a smooth correspondence from improved signature functions. The cost is however a high runtime per pair for the correspondence itself, and the necessity to go through an expensive training stage during preprocessing.

Once correspondence has been established between all meshes in the collection, we carry out a further sanity check by reporting the geodesic distance, or deformation, given by

$$\delta \mathbf{G}_{\mathcal{M}_1\mathcal{M}_2} = \sum_{i,j} \left| \left[\mathbf{G}_{\mathcal{M}_1}\right]_{i,j\in E} - \left[\mathbf{G}_{\mathcal{M}_2}\right]_{\Phi(i),\Phi(j)} \right|, \qquad (19)$$

where E is the set of edges of the meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, and observing how it compares between the different species present in the dataset. We hypothesize that samples from different species will exhibit more shape deviations than individuals from the same species, and investigate whether this can be observed in the geodesic distances retrieved after correspondence established by μMatch. The quantity (Equation 19) is calculated between all pairs in the collection, producing a $n \times n$ matrix $\delta\mathbf{G}$, with n the number of meshes in the collection. Each sub-blocks of the matrix corresponding to the different species can be averaged to produce a reduced distance matrix, depicted in **Figure 7B** for the full μMatch pipeline (with deep functional maps and product manifold filter). The geodesic distances between species match what would be expected from their phylogeny: molar surfaces of chimpanzees and bonobos are highly similar, and also closely resemble human teeth, while gorilla and orangutan molars are observed to be morphologically more dissimilar. In addition to their differences with other species, samples from the orangutan class appear to be extremely diverse, resulting in a large intra-class distance. This experiment provides a sanity check assessing the validity of the algorithms implemented in the μMatch pipeline. We orient the reader interested in a comparative assessment of the relative performance of the selected correspondence algorithms against published alternatives to the original works introducing functional mapping (Ovsjanikov et al., 2012, 2016).

4.2. Case Study: Joint Shape Analysis of Embryonic Limbs and *Dusp6* Gene Expression Patterns

In our previous study (Martínez-Abadías et al., 2018), we performed geometric morphogenesis on a mouse model of Apert syndrome, in which the *Fgfr2* gene contains a mutation which models the human syndrome. The goal was to explore whether shape analysis of a gene expression pattern could suggest the molecular basis of the phenotype. In addition to analyzing the developing anatomical changes in the limb, we also analyzed the 3D expression pattern of *Dusp6*, a gene whose expression reflects the activity of FGF signaling. This allowed subtle alterations in the gene expression pattern to be detected before the anatomical phenotype was apparent, thus strengthening the idea that altered FGF signaling is directly responsible for the phenotype. That previous analysis however depended on manual annotation of 3D landmarks on the data, a process which was very labor-intensive and taking many weeks to achieve. Here, we demonstrate that μMatch can be used to automate this process, jointly quantify the evolution of the shape of embryonic limb and gene expression pattern over developmental time without human bias and achieving similar results in a fraction of the time. We have chosen to reproduce the results of this study to illustrate the use of μMatch on real biological data containing objects that differ in size, orientation,

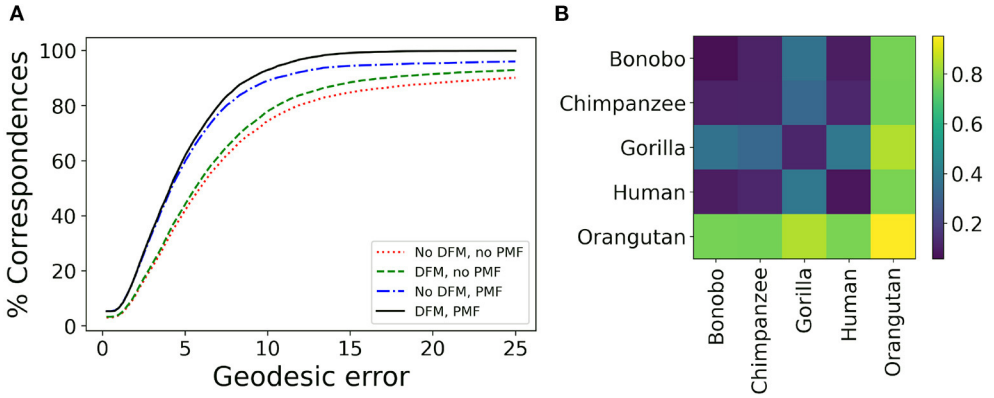


FIGURE 7 | Validation of μMatch. **(A)** Cumulative geodesic error curve, reporting the proportion of predicted correspondence with an error (Equation 18) smaller than a variable threshold, for different setting including or not the deep functional maps (DFM) and product manifold filter (PMF) steps. The complete μMatch pipeline corresponds to the solid black curve (DMF, PMF). **(B)** Matrix of average geodesic distance (Equation 19) between species based on the correspondences obtained with the full μMatch pipeline.

TABLE 3 | Summary of the MOUSE_LIMB dataset.

	Early (∼E10)		Mid (∼E10.5 to ∼E11)		Late (∼E11.5)	
	Limb	***Dusp6***	**Limb**	***Dusp6***	**Limb**	***Dusp6***
Unaffected	12/11	9/5	7/6	6/2	2/3	1/1
Fgfr2 mutant	3/11	6/7	8/8	3/3	7/5	2/5

For each condition, the number of available fore and hind samples of limb surface and Dusp6 gene expression are indicated. Gene expression patterns are not available for all limb samples, hence the smaller sample sizes.

and shape, and exhibit a mixture of inter-group and intra-group variations of morphology.

The dataset we refer to as MOUSE_LIMB, publicly available at dx.doi.org/10.5061/dryad.8h646s0, contains meshes extracted from real optical projection tomography scans (Sharpe et al., 2002) of the limbs and of the *Dusp6* gene expression domains of Apert syndrome (*Fgfr2* mutants) mouse embryos and unaffected littermates ranging from E10 to E11.5, where EN stands for N days after conception. Extracting these meshes requires the raw image data to be first segmented, a step that can be carried out by different methods. We refer readers interested in the details of the segmentation process to the original study (Martínez-Abadías et al., 2018). The meshes are grouped into forelimbs and hind limbs of early (∼E10), mid (∼E10.5 to ∼E11) and late (∼E11.5) developmental periods, according to **Table 3**. To process these data, we set the number of vertices N to be 2000, and use default μMatch values, namely a functional space dimension of $k = 100$, and 100 computed heat kernel and wave kernel signatures.

For each age and genetic background group, a template shape is chosen at random and used to initiate correspondences with the remaining samples. From the mesh correspondences obtained with μMatch, a Procrustes alignment of the limbs surfaces is obtained following the procedure described in Section 3.3.1, yielding an average limb with an associated measure of variance at each vertex. Once limb surfaces are matched, the correspondence can be further used to map internal processes between them, allowing in particular to compare *Dusp6* gene expression data available for some of the limbs. The parameters obtained from the limb alignment can then be used to do a rigid registration of the gene expression patterns. This is then further refined by iterative closest point alignment (Besl and McKay, 1992; Chen and Medioni, 1992) and thin plate spline (TPS) deformation to finely warp the objects onto one another (Duchon, 1977). To implement the TPS deformation, the surfaces are first sub-sampled to obtain M control points $\{c_m\}_{m=1}^{M}$ and the warping is obtained as

$$\delta(x) = \sum_{m=1}^{M} \omega_m \psi(\|x - c_m\|), \tag{20}$$

where $\psi(r) = r^2 \log(r)$ is the thin plate spline kernel. The coefficients $\{\omega_m\}_{m=1}^{M}$ are found by solving for $\mathbf{A}\boldsymbol{\omega} = \mathbf{v}$, where $[\mathbf{A}]_{ij} = \psi(\|c_i - c_j\|) \ \forall \ i,j = 1,\ldots,M$ and $[\mathbf{v}]_m$ is the difference vector between the source and target surface at the control points $m = 1,...,M$. Once the objects have been warped onto one another in this manner, a simple nearest neighbors search is used to obtain a correspondence and the shape analysis procedure described in Section 3.3.1 can be carried out again for the gene expression data. The resulting average limbs and gene expression patterns, color-coded according to local variance, are shown in **Figure 8**. Following Martínez-Abadías et al. (2018), we display results for hind limbs of the early and mid-early stages, and fore limbs of the mid-late and late stages.

In Martínez-Abadías et al. (2018), the limb buds of *Fgfr2* mutants from the late period (∼E11.5) were observed to separate from those of unaffected littermates. During the mid-early (∼E10.5) and early (∼E10) periods, the limbs of *Fgfr2* mutant mice were however undistinguishable from those of unaffected specimen. Significant difference in *Dusp6* gene expression pattern was observed between *Fgfr2* mutant animals and their unaffected littermates for all groups except for the early time period.

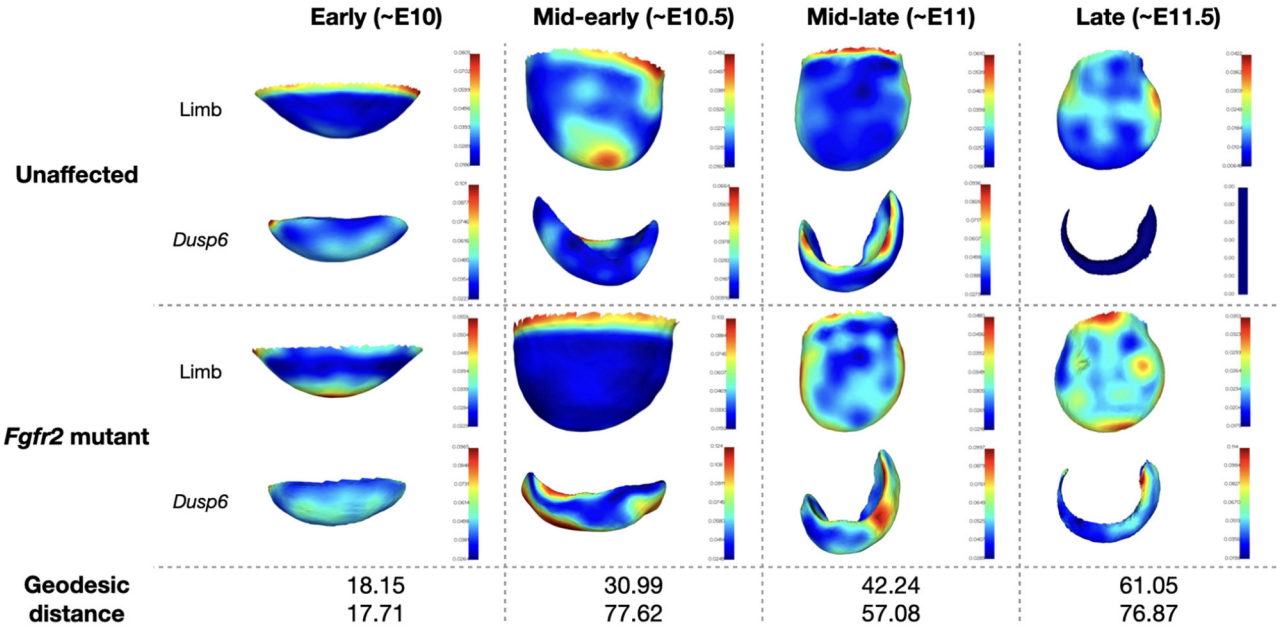


FIGURE 8 | Average limb and *Dusp6* gene expression across the MOUSE_LIMB dataset. The averaged surfaces are color-coded according to the local variance (see colorbars). The geodesic distance between the average shape of the unaffected specimen and *Fgfr2* mutant is also reported for the limb and *Dusp6* gene expression pattern.

We recover the same results using μMatch, as illustrated in **Figure 9**, demonstrating the validity of the automatically retrieved correspondence. The difference in shapes was originally quantified by Procrustes analysis relying on sparse set of semi-landmarks along the most distal edge and on the dorsal and ventral sides of the limb (Martínez-Abadías et al., 2018). In contrast, our approach solves for a dense correspondence, taking into account the entire object's surface. The major advantage of μMicro is that it is entirely landmark-free and fully automated. Once limb surfaces are put in correspondence, their geodesic distance can be used to further quantify the extent of their morphological difference, and the recovered matching used to also align their corresponding gene expression. As such, provided that gene expression patterns have been acquired so as to be appropriately registered inside the limb, comparing the shape of gene expression patterns does neither require artificial landmarking nor a separate correspondence map: they instead get aligned based on their "container." These results demonstrate that μMatch can reliably be used on real bioimage data, which may be noisy, in which samples may exhibit too large variations in their morphology or orientation to rely on volume-based alignment, and which may involve extracted meshes of different numbers of vertices.

5. DISCUSSION AND CONCLUSIONS

We developed μMatch, an automated 3D shape correspondence pipeline tailored to biological objects that are difficult or impossible to landmark. The core element of the pipeline is the correspondence algorithm, which is based on functional mapping. As such, μMatch relies entirely on automatically-extracted signature functions that capture the local geometry of the object's surface to retrieve an optimal matching. In addition to the correspondence algoritm itself, μMatch includes scripts to facilitate the whole workflow, from data pre-processing (including mesh cleaning and pre-computation of important quantities to speed up the correspondence process) to basic shape analysis (including Procrustes analysis and morphing). Since the input data format required by μMatch, namely triangular meshes, is very generic, the pipeline can be used for a broad range of objects. While μMatch has no hard limit on the input mesh size, larger meshes will result in more computationally demanding operations and therefore affect execution time. In our experiments, considered meshes were composed of 2,000 to 3,000 vertices (corresponding to 4,000 to 6,000 faces). For much larger meshes, we recommend to establish a first correspondence at a lower resolution (i.e., on decimated meshes) and then extending it to the full sized meshes. As a reference runtime for standard laptops, establishing correspondence between two meshes of 2,000 vertices took approximately 17 s on a laptop with 16Gb of RAM, an Intel Core i7 8th Generation CPU (8 cores), and an Intel Corporation UHD Graphics 620 (rev 07) GPU. μMatch, is implemented in Python and is freely available on GitHub under BSD-3 Clause license.

We quantitatively validated μMatch relying on the TEETH benchmark dataset (Boyer et al., 2011) for which ground truth correspondence is available. Then, we explored the use

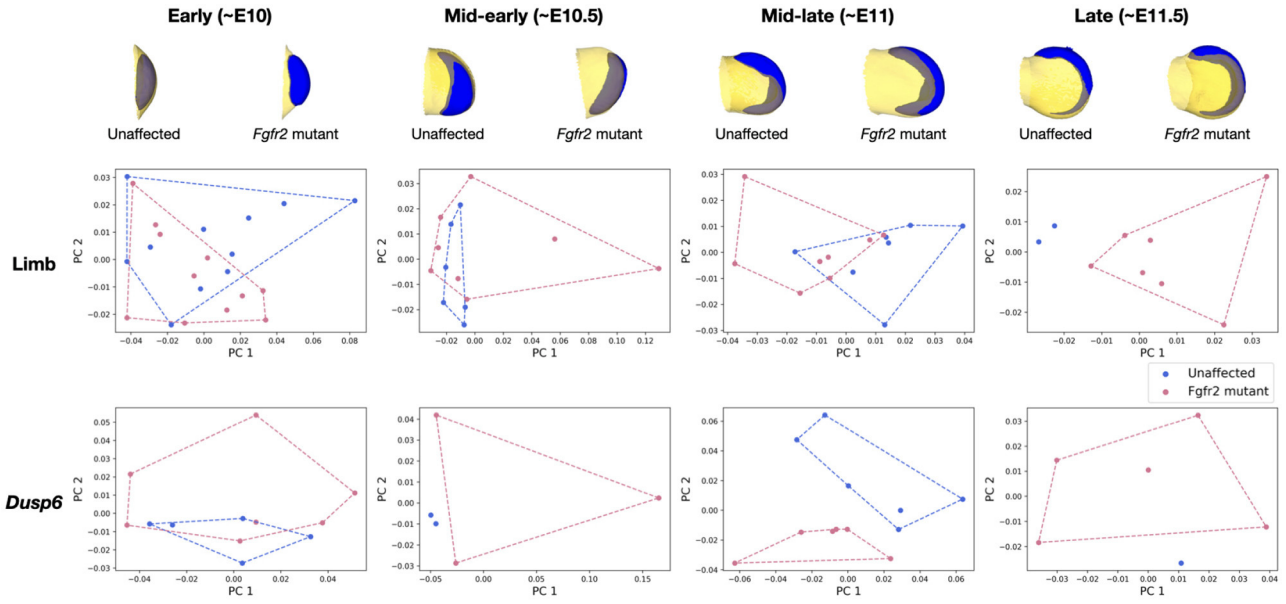


FIGURE 9 | Analysis of the MOUSE_LIMB dataset based on geodesic distances of aligned objects. Vertex-wise deviations from the average shape are concatenated for each mesh in the collection, and reduced by principal component analysis. The resulting two largest principal components are plotted for each sample to quantify the morphological difference between the limbs and *Dusp6* gene expression patterns of *Fgfr2* mutants and unaffected littermates. The average limb (in yellow) and *Dusp6* gene expression pattern (in blue) for each group is displayed on top.

of μMatch to automate the analysis of a dataset of optical projection tomography scans of mouse limbs (MOUSE_LIMB dataset, Martínez-Abadías et al., 2018) and demonstrated that our pipeline allows retrieving published results. This dataset also includes gene expression patterns which morphologies had originally been studied relying on landmarking. We demonstrated that the limb surface correspondence obtained with μMatch can be used to align the corresponding gene expression patterns without the need for any landmarks, and subsequently characterize their morphological differences. Although involving rather small sample sizes, this experiment is a proof-of-principle that the alignments provided by μMatch can be used to compare the spatial morphology of processes that are internal to the surfaces being matched. Beyond allowing to reproduce and automate morphometry from data at the mesoscopic scale, we hope that μMatch will make it possible to investigate new shape-related questions at the cellular or subcellular scale. Being able to reliably put soft-tissue objects without landmarks in correspondence and use their alignment to register internal processes is a first step toward atlas-free registration, which would be useful in many biological studies involving for instance spatial transcriptomics data.

In addition to enabling the quantitative study of collection of equivalent 3D shapes, dense correspondence maps also allow for continuous morphing between meshes. An exciting future direction of this work is therefore to explore how μMatch can be used to build spatiotemporal models of deforming objects. In a biological context, for instance during development or morphogenesis, complexity will almost always significantly increase over time, resulting in possibly challenging correspondence problems between younger, morphologically simpler surfaces and older ones exhibiting more complex shapes. Because correspondence is an essential first step toward mining 3D surface data extracted from bioimages, we hope that μMatch will help enable new avenues in computational morphometry and modeling for biology and lower the entry cost for life scientists intending to rely on statistical shape analysis to explore their data.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: www.wisdom.weizmann.ac.il/~ylipman/CPsurfcomp/; dx.doi.org/10.5061/dryad.8h646s0.

AUTHOR CONTRIBUTIONS

JK and VU contributed to the conception and design of the work, implemented the methods, carried out the experiments, and wrote the first draft of the manuscript. GD, NM-A, and JS prepared the mouse embryo limb dataset and contributed to the experiments. All authors contributed to final manuscript editing, read, and approved the submitted version.

FUNDING

JK, JS, and VU are supported by EMBL internal funding. NM-A was supported by the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement Marie Curie Fellowship FP7-PEOPLE-2012-IIF 327382 and by the Grup de

Recerca Consolidat en Antropologia Biològica (2017 SGR 1630). GD was supported by the ERC Advanced Grant SIMBIONT (670555) and the Spanish Plan Estatal project: LIMBNET-3D PID2019-110868GB-I00.

ACKNOWLEDGMENTS

JK and VU acknowledge the NVIDIA Corporation for the donation of a Titan V GPU that made this research possible.

REFERENCES

Attene, M. (2010). A lightweight approach to repairing digitized polygon meshes. *Vis. Comput*. 26, 1393–1406. doi: 10.1007/s00371-010-0416-3

Aubry, M., Schlickewei, U., and Cremers, D. (2011). "The wave kernel signature: a quantum mechanical approach to shape analysis," in *IEEE International Conference on Computer Vision Workshops* (Barcelona: IEEE), 1626–1633.

Belay, B., Koivisto, J., Parraga, J., Koskela, O., Montonen, T., Kellomäki, M., et al. (2021). Optical projection tomography as a quantitative tool for analysis of cell morphology and density in 3d hydrogels. *Sci. Rep*. 11, 1–10. doi: 10.1038/s41598-021-85996-8

Besl, P., and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell*. 14, 239–256. doi: 10.1109/34.121791

Boehm, B., Rautschka, M., Quintana, L., Raspopovic, J., Jan, ɜ Z., and Sharpe, J. (2011). A landmark-free morphometric staging system for the mouse limb bud. *Development* 138, 1227–1234. doi: 10.1242/dev.057547

Bône, A., Louis, M., Martin, B., and Durrleman, S. (2018). "Deformetrica 4: an open-source software for statistical shape analysis," in *International Workshop on Shape in Medical Imaging* (Granada), 3–13.

Bookstein, F. L. (1997). *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambridge: Cambridge University Press.

Boyer, D., Lipman, Y., Clair, E., Puente, J., Patel, B., Funkhouser, T., et al. (2011). Algorithms to automatically quantify the geometric similarity of anatomical surfaces. *Proc. Natl. Acad. Sci. U.S.A*. 108, 18221–18226. doi: 10.1073/pnas.1112822108

Brodoehl, S., Gaser, C., Dahnke, R., Witte, O., and Klingner, C. (2020). Surface-based analysis increases the specificity of cortical activation patterns and connectivity results. *Sci. Rep*. 10, 1–13. doi: 10.1038/s41598-020-62832-z

Chen, Y., and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image Vis. Comput*. 10, 145–155. doi: 10.1016/0262-8856(92)90066-C

Crouse, D. (2016). On implementing 2d rectangular assignment algorithms. *IEEE Trans. Aerosp Electron. Syst*. 52, 1679–1696. doi: 10.1109/TAES.2016.140952

Driscoll, M., Welf, E., Jamieson, A., Dean, K., Isogai, T., Fiolka, R., et al. (2019). Robust and automated detection of subcellular morphological motifs in 3d microscopy images. *Nat. Methods* 16, 1037–1044. doi: 10.1038/s41592-019-0539-z

Dryden, I., and Mardia, K. (2016). *Statistical Shape Analysis: With Applications in R, Vol. 995*. Hoboken, NJ: John Wiley & Sons.

Duchon, J. (1977). "Splines minimizing rotation-invariant semi-norms in sobolev spaces," in *Constructive Theory of Functions of Several Variables* (Oberwolfach), 85–100.

Finka, L., Luna, S., Brondani, J., Tzimiropoulos, Y., McDonagh, J., Farnworth, M., et al. (2019). Geometric morphometrics for the study of facial expressions in non-human animals, using the domestic cat as an exemplar. *Sci. Rep*. 9, 1–12. doi: 10.1038/s41598-019-46330-5

Gower, J., and Dijksterhuis, G. (2004). *Procrustes Problems, Vol. 30*. Oxford: Oxford University Press.

Grocott, T., Thomas, P., and Münsterberg, A. (2016). Atlas toolkit: fast registration of 3d morphological datasets in the absence of landmarks. *Sci. Rep*. 6, 1–7. doi: 10.1038/srep20732

Hahn, M., Nord, C., Franklin, O., Alanentalo, T., Mettävainio, M., Morini, F., et al. (2020). Mesoscopic 3d imaging of pancreatic cancer and langerhans islets based on tissue autofluorescence. *Sci. Rep*. 10, 1–11. doi: 10.1038/s41598-020-74616-6

Halimi, O., Litany, O., Rodola, E., Bronstein, A., and Kimmel, R. (2019). "Unsupervised learning of dense shape correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4370–4379.

Heinrich, L., Bennett, D., Ackerman, D., Park, W., Bogovic, J., Eckstein, N., et al. (2020). Automatic whole cell organelle segmentation in volumetric electron microscopy. *bioRxiv*. doi: 10.1101/2020.11.14.382143

Horstmann, M., Topham, A., Stamm, P., Kruppert, S., Colbourne, J., Tollrian, R., et al. (2018). Scan, extract, wrap, compute a 3d method to analyse morphological shape differences. *PeerJ*. 6:e4861. doi: 10.7717/peerj.4861

Isensee, F., Jaeger, P., Kohl, S., Petersen, J., and Maier-Hein, K. (2021). nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat. Methods* 18, 203–211. doi: 10.1038/s41592-020-01008-z

Jacobson, A., and Panozzo, D. (2018). *Libigl: A Simple C++ Geometry Processing Library*. Available online at: https://libigl.github.io/.

Kalinin, A., Allyn-Feuer, A., Ade, A., Fon, G.-V., Meixner, W., Dilworth, D., et al. (2018). "3d cell nuclear morphology: microscopy imaging dataset and voxel-based morphometry classification results," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (Salt Lake City, UT: IEEE), 2272–2280.

Karanovic, T., and Bláha, M. (2019). Taming extreme morphological variability through coupling of molecular phylogeny and quantitative phenotype analysis as a new avenue for taxonomy. *Sci. Rep*. 9, 1–15. doi: 10.1038/s41598-019-38875-2

Kendall, D. (1989). A survey of the statistical theory of shape. *Stat. Sci*. 4, 87–99. doi: 10.1214/ss/1177012582

Kilian, M., Mitra, N., and Pottmann, H. (2007). "Geometric modeling in shape space," in *ACM SIGGRAPH 2007* (San Diego, CA), 64-es.

Kim, V., Lipman, Y., and Funkhouser, T. (2011). Blended intrinsic maps. *ACM Trans. Graph*. 30, 1–12. doi: 10.1145/2070781.2024224

Klein, S., Staring, M., Murphy, K., Viergever, M., and Pluim, J. (2009). Elastix: a toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging* 29, 196–205. doi: 10.1109/TMI.2009.2035616

Klingenberg, C. (2008). Novelty and homology-freeİ morphometrics: what's in a name? *Evol. Biol*. 35, 186–190. doi: 10.1007/s11692-008-9029-4

Koehl, P., and Hass, J. (2015). Landmark-free geometric methods in biological shape analysis. *J. R. Soc. Interface* 12, 20150795. doi: 10.1098/rsif.2015.0795

Laga, H., Kurtek, S., Srivastava, A., and Miklavcic, S. (2014). Landmark-free statistical analysis of the shape of plant leaves. *J. Theor. Biol*. 363:41–52. doi: 10.1016/j.jtbi.2014.07.036

Lipman, Y., and Funkhouser, T. (2009). Möbius voting for surface correspondence. *ACM Trans. Graph*. 28, 1–12. doi: 10.1145/1531326.1531378

Litany, O., Remez, T., Rodol, E., Bronstein, A., and Bronstein, M. (2017). "Deep functional maps: structured prediction for dense shape correspondence," in *IEEE International Conference on Computer Vision* (Venice: IEEE), 5659–5667.

Lorensen, W., and Cline, H. (1987). "Marching cubes: a high resolution 3d surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA), 163–169.

Lucas, A., Ryder, P., Li, B., Cimini, B., Eliceiri, K., and Carpenter, A. (2021). Open-source deep-learning software for bioimage segmentation. *Mol. Biol. Cell*. 32, 823–829. doi: 10.1091/mbc.E20-10-0660

Lucas, M., Kenobi, K., Von Wangenheim, D., Voβ,, U., Swarup, K., De Smet, I., et al. (2013). Lateral root morphogenesis is dependent on the mechanical properties of the overlaying tissues. *Proc. Natl. Acad. Sci. U.S.A*. 110, 5229–5234. doi: 10.1073/pnas.1210807110

Martínez-Abadías, N., Esparza, M., Sjøvold, T., González-José, R., Santos, M., Hernández, M., et al. (2012). Pervasive genetic integration directs the evolution of human skull shape. *Evolution* 66, 1010–1023. doi: 10.1111/j.1558-5646.2011.01496.x

Martínez-Abadías, N., Estivill, R., Tomas, J., Perrine, S., Yoon, M., Robert-Moreno, A., et al. (2018). Quantification of gene expression patterns to reveal the origins of abnormal morphogenesis. *Elife* 7:e36405. doi: 10.7554/eLife.36405

Melzi, S., Ren, J., Rodolà, K. E., Sharma, A., Ovsjanikov, M., and Wonka, P. (2019). ZoomOut: spectral upsampling for efficient shape correspondence. *ACM Trans. Graph*. 38, 1–14. doi: 10.1145/3355089.3356524

Miao, S., Wang, Z., and Liao, R. (2016). A cnn regression approach for real-time 2d/3d registration. *IEEE Trans. Med. Imaging* 35, 1352–1363. doi: 10.1109/TMI.2016.2521800

Mitchell, J., Mount, D., and Papadimitriou, C. (1985). The discrete geodesic problem. *SIAM J. Comput*. 16, 647–668. doi: 10.1137/0216045

Musy, M., Jacquenot, G., Dalmasso, G., Neoglez., de Bruin, R., Pollack, A., et al. (2021). vedo, a python module for scientific analysis and visualization of 3D objects and point clouds. *Zenodo*. doi: 10.5281/zenodo.4287635

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L. (2012). Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph*. (Macao) 31, 1–11. doi: 10.1145/2185520.2185526

Ovsjanikov, M., Corman, E., Bronstein, M., Rodolà, E., Ben-Chen, M., Guibas, L., et al. (2016). "Computing and processing correspondences with functional maps," in *ACM SIGGRAPH Asia 2016 Courses*, 1–60.

Paul-Gilloteaux, P., Heiligenstein, X., Belle, M., Domart, M.-C., Larijani, B., Collinson, L., et al. (2017). ec-clem: flexible multidimensional registration software for correlative microscopies. *Nat. Methods* 14, 102–103. doi: 10.1038/nmeth.4170

Phillip, J., Han, K.-S., Chen, W.-C., Wirtz, D., and Wu, P.-H. (2021). A robust unsupervised machine-learning method to quantify the morphological heterogeneity of cells and nuclei. *Nat. Protoc*. 16, 754–774. doi: 10.1038/s41596-020-00432-x

Pinkall, U., and Polthier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Exp. Math*. 2, 15–36. doi: 10.1080/10586458.1993.10504266

Preibisch, S., Saalfeld, S., Schindelin, J., and Tomancak, P. (2010). Software for bead-based registration of selective plane illumination microscopy data. *Nat. Methods* 7, 418–419. doi: 10.1038/nmeth0610-418

Ramirez, P., Zammit, J., Vanderpoorten, O., Riche, F., Blé, F.-X., Zhou, X.-H., et al. (2019). Optij: open-source optical projection tomography of large organ samples. *Sci. Rep*. 9, 1–9. doi: 10.1038/s41598-019-52065-0

Rohlf, F., and Slice, D. (1990). Extensions of the procrustes method for the optimal superimposition of landmarks. *Syst. Biol*. 39, 40–59. doi: 10.2307/2992207

Schmidt, P., Born, J., Campen, M., and Kobbelt, L. (2019). Distortion-minimizing injective maps between surfaces. *ACM Trans. Graph*. 38, 1–15. doi: 10.1145/3355089.3356519

Schreiner, J., Asirvatham, A., Praun, E., and Hoppe, H. (2004). "Inter-surface mapping," in *ACM SIGGRAPH 2004* (Los Angeles, CA).

Sharpe, J., Ahlgren, U., Perry, P., Hill, B., Ross, A., Hecksher-Sørensen, J., et al. (2002). Optical projection tomography as a tool for 3d microscopy and gene expression studies. *Science* 296, 541–545. doi: 10.1126/science.1068206

Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. (2004). "The princeton shape benchmark," in *Proceedings of the International Conference on Shape Modeling Applications* (Genova), 167–178.

Sorkine, O., and Alexa, M. (2007). "As-rigid-as-possible surface modeling," in *ACM SIGGRAPH 2007* (San Diego, CA), 109–116.

Srivastava, A., Klassen, E., Joshi, S., and Jermyn, I. (2010). Shape analysis of elastic curves in euclidean spaces. *IEEE Trans. Pattern Anal. Mach. Intell*. 33, 1415–1428. doi: 10.1109/TPAMI.2010.184

Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum* 28, 1383–1392. doi: 10.1111/j.1467-8659.2009.01515.x

Toussaint, N., Redhead, Y., Vidal-García, M., Lo Vercio, L., Liu, W., Fisher, E., et al. (2021). A landmark-free morphometrics pipeline for high-resolution phenotyping: application to a mouse model of down syndrome. *Development* 148, dev188631. doi: 10.1242/dev.188631

van der Walt, S., Schönberger, J., Nunez-Iglesias, J., Boulogne, F., Warner, J., Yager, N., et al. (2014). scikit-image: image processing in Python. *PeerJ*. 2:e453. doi: 10.7717/peerj.453

Van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D. (2011). A survey on shape correspondence. *Comput. Graph. Forum* 30, 1681–1707. doi: 10.1111/j.1467-8659.2011.01884.x

Vestner, M., Litman, R., Rodol, E., Bronstein, A., and Cremers, D. (2017). "Product manifold filter: non-rigid shape correspondence via kernel density estimation in the product space," in *IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 6681–6690.

Voigt, F., Kirschenbaum, D., Platonova, E., Pagès, S., Campbell, R., Kastli, R., et al. (2019). The mesospim initiative: open-source light-sheet microscopes for imaging cleared tissue. *Nat. Methods* 16, 1105–1108. doi: 10.1038/s41592-019-0554-0

White, J., Ortega-Castrillón, A., Matthews, H., Zaidi, A., Ekrami, O., Snyders, J., et al. (2019). Meshmonk: open-source large-scale intensive 3d phenotyping. *Sci. Rep*. 9, 1–11. doi: 10.1038/s41598-019-42533-y

Windheuser, T., Schlickewei, U., Schmidt, F., and Cremers, D. (2011). "Geometrically consistent elastic matching of 3d shapes: a linear programming solution," in *IEEE International Conference on Computer Vision* (Barcelona: IEEE), 2134–2141.

Yang, X., Kwitt, R., Styner, M., and Niethammer, M. (2017). "Fast predictive multimodal image registration," in *IEEE 14th International Symposium on Biomedical Imaging* (Melbourne, VIC: IEEE), 858–862.

Yeh, Y.-C., Weng, C.-H., Huang, Y.-J., Fu, C.-J., Tsai, T.-T., and Yeh, C.-Y. (2021). Deep learning approach for automatic landmark detection and alignment analysis in whole-spine lateral radiographs. *Sci. Rep*. 11, 1–15. doi: 10.1038/s41598-021-87141-x

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 Computer Science

ORIGINAL RESEARCH
published: 03 March 2022
doi: 10.3389/fcomp.2022.805166

From Shallow to Deep: Exploiting Feature-Based Classifiers for Domain Adaptation in Semantic Segmentation

Alex Matskevych, Adrian Wolny, Constantin Pape and Anna Kreshuk**

Cell Biology and Biophysics Unit, European Molecular Biology Laboratory, Heidelberg, Germany

The remarkable performance of Convolutional Neural Networks on image segmentation tasks comes at the cost of a large amount of pixelwise annotated images that have to be segmented for training. In contrast, feature-based learning methods, such as the Random Forest, require little training data, but rarely reach the segmentation accuracy of CNNs. This work bridges the two approaches in a transfer learning setting. We show that a CNN can be trained to correct the errors of the Random Forest in the source domain and then be applied to correct such errors in the target domain without retraining, as the domain shift between the Random Forest predictions is much smaller than between the raw data. By leveraging a few brushstrokes as annotations in the target domain, the method can deliver segmentations that are sufficiently accurate to act as pseudo-labels for target-domain CNN training. We demonstrate the performance of the method on several datasets with the challenging tasks of mitochondria, membrane and nuclear segmentation. It yields excellent performance compared to microscopy domain adaptation baselines, especially when a significant domain shift is involved.

Keywords: microscopy segmentation, domain adaptation, deep learning, transfer learning, biomedical segmentation

OPEN ACCESS

Edited by:
Florian Jug,
Human Technopole, Italy

Reviewed by:
Dagmar Kainmueller,
Max Delbrück Center for Molecular Medicine, Helmholtz Association of German Research Centers (HZ), Germany
Stavros Tsogkas,
Samsung AI Center Toronto, Canada

****Correspondence:***
Constantin Pape
constantin.pape@embl.de
Anna Kreshuk
anna.kreshuk@embl.de

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *29 October 2021*
Accepted: *02 February 2022*
Published: *03 March 2022*

Citation:
Matskevych A, Wolny A, Pape C and Kreshuk A (2022) From Shallow to Deep: Exploiting Feature-Based Classifiers for Domain Adaptation in Semantic Segmentation. Front. Comput. Sci. 4:805166. doi: 10.3389/fcomp.2022.805166

1. INTRODUCTION

Semantic segmentation—partitioning the image into areas of biological (semantic) meaning—is a ubiquitous problem in microscopy image analysis. Compared to natural images, microscopy segmentation problems are particularly well-suited for feature-based ("shallow") machine learning, as the difference between semantic classes can often be captured in local edge, texture, or intensity descriptors (Belevich et al., 2016; Arganda-Carreras et al., 2017; Berg et al., 2019). While convolutional neural networks (CNNs) have long overtaken feature-based approaches in segmentation accuracy and inference speed, interactive feature-based solutions continue to attract users due to the low requirements to training data volumes, nearly real-time training speeds and general simplicity of the setup, which does not require computational expertise.

CNNs are made up of millions of learnable parameters which have to be configured based on user-provided training examples. With insufficient training data, CNNs are very prone to overfitting, "memorizing" the training data instead of deriving generalizable rules. Strategies to suppress overfitting include data augmentation (Ronneberger et al., 2015), incorporation of prior information (El Jurdi et al., 2021), dropout and sub-network re-initialization (Han et al., 2016; Taha et al., 2021) and, in case a similar task has already been solved on sufficiently similar data, domain

adaptation, and transfer learning. In the latter case, the network exploits a large amount of labels in the so called "source" domain to learn good parameter values for the task at hand, which are further adapted for the unlabeled or sparsely labeled "target" domain through unsupervised or weakly supervised learning. For microscopy images, the adaptation is commonly achieved by bringing the distributions of the source and target domain data closer to each other, either by forcing the network to learn domain-invariant features (Long et al., 2015; Roels et al., 2019; Liu et al., 2020) or by using generative networks and cycle consistency constraints (Zhang et al., 2018; Chen et al., 2019; Januszewski and Jain, 2019). Alternatively, the domain shift can be explicitly learned in a part of the network (Rozantsev et al., 2018). In addition to labels in the source domain, pseudo-labels in the target domain are often used for training (Choi et al., 2019; Xing et al., 2019). Pseudo-labels can be computed from the predictions of the source domain network (Choi et al., 2019) or predictions for pixels similar to source domain labels (Bermúdez-Chacón et al., 2019).

In contrast, Random Forest (RF), one of the most popular "shallow" learning classifiers (Breiman, 2001), does not overfit on small amounts of training data and trains so fast that in practice no domain adaptation strategies are applied—the classifier is instead fully retrained with sparse labels in the target domain. However, unlike a CNN, it cannot fully profit from large amounts of training data. The aim of our contribution is to combine the best of both worlds, exploiting fast training of the Random Forest for domain adaptation and excellent performance of CNNs for accurate segmentation with large amounts of training data. We use the densely labeled source domain to train many Random Forests for segmentation and then train a CNN for Random Forest prediction enhancement (see **Figure 1**). On the target domain, we train a new Random Forest from a few brushstroke labels and simply apply the pre-trained Prediction Enhancer (PE) network to improve the probability maps. The enhanced predictions are substantially more accurate than the Random Forest or a segmentation CNN trained only on the source domain. Furthermore, a new CNN can be trained using enhanced predictions as pseudo-labels, achieving an even better accuracy with no additional annotation cost. Since the Prediction Enhancer is only trained on RF probability maps, it remains agnostic to the appearance of the raw data and can therefore be applied to mitigate even very large domain gaps between source and target datasets, as long as the segmentation task itself remains similar. To illustrate the power of our approach, we demonstrate domain adaptation between different datasets of the same modality, and also from confocal to light sheet microscopy, from electron to confocal microscopy and from fluorescent light microscopy to histology. From the user perspective, domain adaptation is realized in a straightforward, user-friendly setting of training a regular U-Net, without adversarial elements or task re-weighting. Furthermore, a well-trained Prediction Enhancer network can be used without retraining, only requiring training of the Random Forest from the user. Our Prediction Enhancer

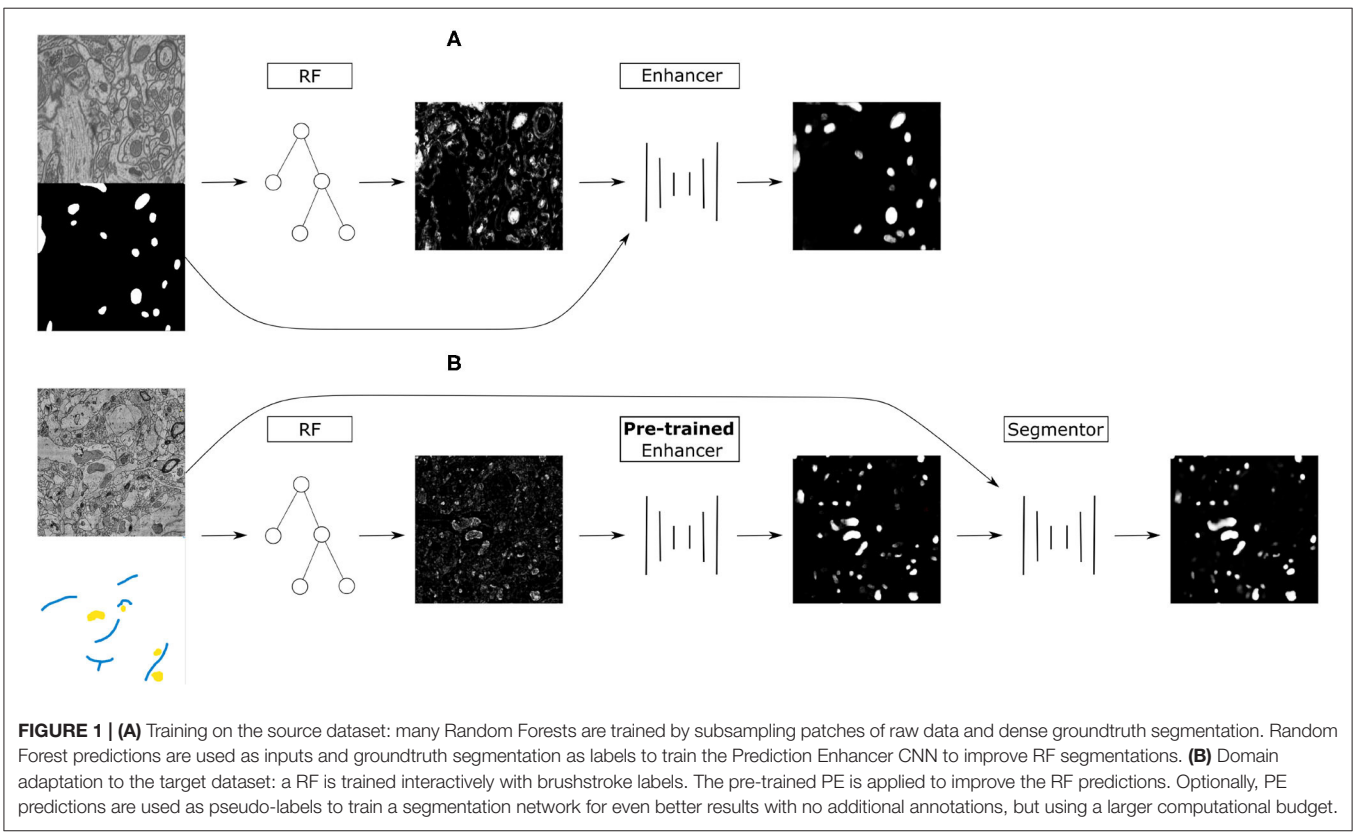


FIGURE 1 | (A) Training on the source dataset: many Random Forests are trained by subsampling patches of raw data and dense groundtruth segmentation. Random Forest predictions are used as inputs and groundtruth segmentation as labels to train the Prediction Enhancer CNN to improve RF segmentations. **(B)** Domain adaptation to the target dataset: a RF is trained interactively with brushstroke labels. The pre-trained PE is applied to improve the RF predictions. Optionally, PE predictions are used as pseudo-labels to train a segmentation network for even better results with no additional annotations, but using a larger computational budget.

networks for mitochondria, nuclei, or membrane segmentation tasks are available at the BioImage Model Zoo (https://bioimage.io) and can easily be applied to improve predictions of the Pixel Classification workflow in ilastik or of the Weka Trainable Segmentation plugin in Fiji.

2. METHODS

Our approach combines the advantages of feature-based and end-to-end segmentation methods by training a Prediction Enhancer network to predict one from the other. On the target dataset, retraining can be limited to the feature-based classifier as its predictions—unlike the raw data—do not exhibit a significant domain shift if the same semantic classes are being segmented. In more detail, we propose the following sequence of steps (see also **Figure 1**):

1. Create training data for the Prediction Enhancer CNN by training multiple Random Forests on random samples of the densely labeled source domain.
2. Train the Prediction Enhancer using the RF predictions as input and the ground-truth segmentation as labels.
3. Train a Random Forest on the target dataset with a few brushstroke labels and use the pre-trained Prediction Enhancer to improve the predictions.
4. Use the improved predictions as pseudo-labels to train a CNN on the target dataset. This step is optional and trades improved quality for the computational cost of training a CNN from scratch.

Note that the Prediction Enhancer only takes the predictions of the Random Forest as input. Neither raw data nor labels of the source dataset are needed to apply it to new data. Our method can therefore be classified as *source-free domain adaption*, but the additional feature-based learning step allows us to avoid training set estimation or reconstruction, commonly used in other source-free or knowledge distillation-based approaches like Du et al. (2021) and Liu et al. (2021). At the same time, we can fully profit from all advances in the field of pseudo-label rectification (Prabhu et al., 2021; Wu et al., 2021; Zhang et al., 2021; Zhao et al., 2021), applying those to pseudo-labels generated by the PE network.

2.1. Prediction Enhancer

The Prediction Enhancer is based on the U-Net architecture (Ronneberger et al., 2015). To create training data, we train multiple Random Forests on the dense labels of the source domain, using the same pixel features as in the ilastik pixel classification workflow (Berg et al., 2019). To obtain a diverse set of shallow classifiers we sample patches of various size and train a classifier for each patch based on the raw data and dense labels. Typically, we train 500–1,000 different classifiers. Next, we train the U-Net following the standard approach for semantic segmentation, using Random Forest predictions (but not the raw data) as input and the provided dense labels of the source domain as the groundtruth. To create more variability, we sample from all previously trained classifiers. We use either the binary cross entropy or the Dice score as loss function.

Segmentation of a new dataset only requires training a single Random Forest; its predictions can directly be improved with the pre-trained Prediction Enhancer. Here, we use ilastik pixel classification workflow, which enables training a Random Forest interactively from brushstroke user annotations.

2.2. Further Domain Adaptation With Pseudo-Labels

The Prediction Enhancer can improve the segmentation results significantly, as shown in Section 3. However, it relies only on the Random Forest predictions, and can thus not take intensity, texture or other raw image information into account. To make use of such information and further improve segmentation results, we can use the predictions of the Enhancer as pseudo-labels and train a segmentation U-Net on the target dataset. We use either Dice score or binary cross entropy as loss and make the following adjustments to the standard training procedure to enable training from noisy pseudo-labels:

- Use the RF predictions as soft labels in range $[0, 1]$ instead of hard labels in $\{0, 1\}$.
- Use a simple label rectification strategy to weight the per-pixel loss based on the prediction confidence (see Section 2.2.1).
- In the final loss, add a consistency term similar to Tarvainen and Valpola (2017) that compares the current predictions to the predictions of the network's exponential moving average (see Section 2.2.2).

2.2.1. Label Rectification

Label rectification is a common strategy in self-learning-based domain adaptation methods, where predictions from the source model are used as pseudo-labels on the target domain. Rectification is then used to correct for the label noise. Several strategies have been proposed, for example based on the distance to class prototypes in the feature space (Zhang et al., 2021) or prediction confidence after several rounds of dropout (Wu et al., 2021).

Here, we adopt a simple label rectification strategy based on the prediction confidence to weight the pseudo-labels y:

$$\hat{y}_k = \omega_k \, y_k, \tag{1}$$

where k is the class index. The pseudo labels y_k correspond to the predictions of the Prediction Enhancer and are continuous in the range $[0, 1]$. For the case of foreground/background segmentation $k \in \{0, 1\}$ and we define the per-pixel weight for the foreground class as

$$\omega_1 = 1 - \mathrm{abs}(p_1 - \eta_1). \tag{2}$$

Here, p_1 is the foreground probability map predicted by the segmentation network and η_1 a scalar value, defined as the exponentially weighted average computed over the foreground mask S:

$$\begin{aligned} \eta_1 &\leftarrow \lambda \, \eta_1 + (1 - \lambda) \ast \mathrm{mean}(S), \\ &\text{where } S = \{p_1(x) | x \in X \text{ and } y_1(x) > 0.5\}. \end{aligned} \tag{3}$$

Here, X is the set of all pixels in the input image. We set $\lambda = 0.999$ in all experiments. The weight ω_0 for the background class is computed in the same manner.

2.2.2. Consistency Loss Term

For training with pseudo-labels we introduce a consistency term in the loss function, which is based on the "Mean Teacher" training procedure for semi-supervised classification (Tarvainen and Valpola, 2017). The loss term compares the output of the network f with the output of the network g, defined as the exponential moving average (EMA) of f. This method promotes more consistent predictions across training iterations. We make use of this method for training a segmentation network (parameterized by θ_f) from pseudo-labels. Its EMA, g is parameterized by

$$\theta_g \leftarrow \alpha\theta_g + (1-\alpha)\theta_f, \tag{4}$$

where we set the smoothing coefficient α to 0.999 following (Tarvainen and Valpola, 2017).

Given that we are comparing the per pixel predictions of the current network and its EMA, we use the loss function that is also employed for comparing to the pseudo labels: we either use the Dice loss

$$L_{Dice,c}(p_f, p_g) = \frac{2\sum_i^N p_{f,i}\, p_{g,i}}{\sum_i^N p_{f,i}^2 + \sum_i^N p_{g,i}^2} \tag{5}$$

or the binary cross entropy loss

$$L_{BCE,c}(p_f, p_g) = \frac{1}{N}\sum_i^N p_{g,i} log(p_{f,i}) + (1-p_{g,i})(1-log(p_{f,i})). \tag{6}$$

Here x denotes the input image, $p_f = f(x)$, $p_g = g(x)$, and N is the number of pixels. The combined loss function is

$$L_R^{full} = L_R + L_{R,c}, \tag{7}$$

where R is either *Dice* or *BCE*. The term L_R compares the output from f with pseudo-labels defined in Equation 1 and $L_{R,c}$ is the consistency term.

3. RESULTS

3.1. Data and Setup

We evaluate the proposed domain adaptation method on challenging semantic segmentation problems, including mitochondria segmentation in Electron Microscopy (EM), membrane segmentation in electron, and light microscopy (LM) as well as nucleus segmentation in LM. **Table 1** summarizes all datasets used for the experiments. **Table A1** lists the data size as well as the train, validation, and test splits for all datasets.

Some of the datasets we use represent image stacks and could be processed as 3D volumes with different levels of anisotropy. We choose to process them as independent 2D images instead to enable a wider set of source/target domain pairs. If not noted otherwise, training from pseudo-labels is performed using the consistency loss term and label rectification (Equation 7). We use a 2D U-Net architecture (Ronneberger et al., 2015) with 64 features in the initial layer, four downsampling/upsampling levels and double the number of features per level for all networks. The network and training code is based on the PyTorch implementation from Wolny et al. (2020). For all training runs we use the Adam optimizer with initial learning rate of 0.0002, weight decay of 0.00001. Furthermore, we decrease the learning rate by a factor of 0.2 if the validation metric is not improving for a dataset dependent number of iterations. We use binary cross entropy as a loss function for the mitochondria (Section 3.2) and nucleus (Section 3.4) segmentation and dice loss for the membrane segmentation (Section 3.3).

3.2. Mitochondria Segmentation

We first perform mitochondria segmentation in EM. We train the Prediction Enhancer on the EPFL dataset (the only FIB/SEM dataset in the collection) and then perform source-free domain

TABLE 1 | The datasets used in the experiments.

Name	EPFL	VNC	MitoEM-R	MitoEM-H	Kasthuri	CREMI
(A) ELECTRON MICROSCOPY DATASETS USED IN THE EXPERIMENTS.						
Organism/tissue	Mouse/hippocampus	Fruitfly/ventral nerve cord	Rat/cortex	Human/cortex	Mouse/cortex	Fruitfly/brain
Modality	FIBSEM	ssTEM	sbEM	sbEM	ssTEM	ssTEM
Tasks	Mitochondria	Mitochondria, membranes	Mitochondria	Mitochondria	Mitochondria	Membranes
Resolution	5 × 5 × 5 nm	45 × 5 × 5 nm	30 × 8 × 8 nm	30 × 8 × 8 nm	30 × 3 × 3 nm	40 × 4 × 4 nm
References	Lucchi et al., 2013	Gerhard et al., 2013	Wei et al., 2020	Wei et al., 2020	Kasthuri et al., 2015	cremi.org

Name	Root	Ovules	DSB-FL	Monuseg
(B) LIGHT MICROSCOPY DATASETS USED IN THE EXPERIMENTS.				
Organism/tissue	Arabidopsis/lateral root	Arabidopsis/ovules	Various/nuclear stain	Human/kidney
Modality	Lightsheet	Confocal	Fluorescence	Histopathology
Tasks	Membranes	Membranes	Nuclei	Nuclei
Resolution	0.25×0.1625×0.1625 μm	0.235×0.075×0.075 μm		
References	Wolny et al., 2020	Wolny et al., 2020	Caicedo et al., 2019	Kumar et al., 2019

adaptation on the VNC, MitoEM-R, MitoEM-H, and Kasthuri datasets. For domain adaption, the Random Forest for initial target prediction is trained interactively in ilastik using a separate train split. The RF predictions are then improved by the PE and the improved predictions are used as pseudo-labels for a U-Net trained from scratch (Pseudo-label Net). We compare to direct predictions of a U-Net trained for Mitochondria segmentation on the source domain EPFL (Source Net) and to the Y-Net (Roels et al., 2019), a different method for domain adaptation, which is unsupervised on the target domain, but not source-free. We also indicate the performance of a U-Net trained on the target dataset as an estimate of the upper bound of the achievable performance (a separate train split is used).

TABLE 2 | Results for mitochondria segmentation in EM.

Model/Dataset	EPFL	VNC	MitoEM-R	MitoEM-H	Kasthuri
Source net	0.933	0.695	0.738	0.591	0.723
Y-Net	–	0.713	0.781	0.678	0.0
RF	0.625	0.647	0.511	0.338	0.590
PE	0.824	0.840	0.705	0.624	0.778
Pseudo-label net	–	**0.884**	**0.793**	**0.751**	**0.834**
Target net	0.933	0.891	0.939	0.920	0.942

Quality is measured by the F1-score of the mitochondria prediction (higher is better). EPFL dataset is used as the source for domain adaptation by the Y-Net, Prediction Enhancer (PE), and Pseudo-label net. Best result is shown in bold.

Table 2 summarizes the resulting F1 scores (higher is better) for the source dataset and all target datasets. The Enhancer improves the Random Forest predictions significantly on all target datasets and the CNN trained from pseudo-labels further improves the results. The pseudo-label CNN always performs better than the source network or the Y-Net, which fails completely for the Kasthuri dataset where the domain gap is particularly large. **Figure 2** shows an example of the improvements from RF to PE and PE to Pseudo-label Net.

For the mitochondria segmentation task we also check if training the PE on multiple source datasets improves results.

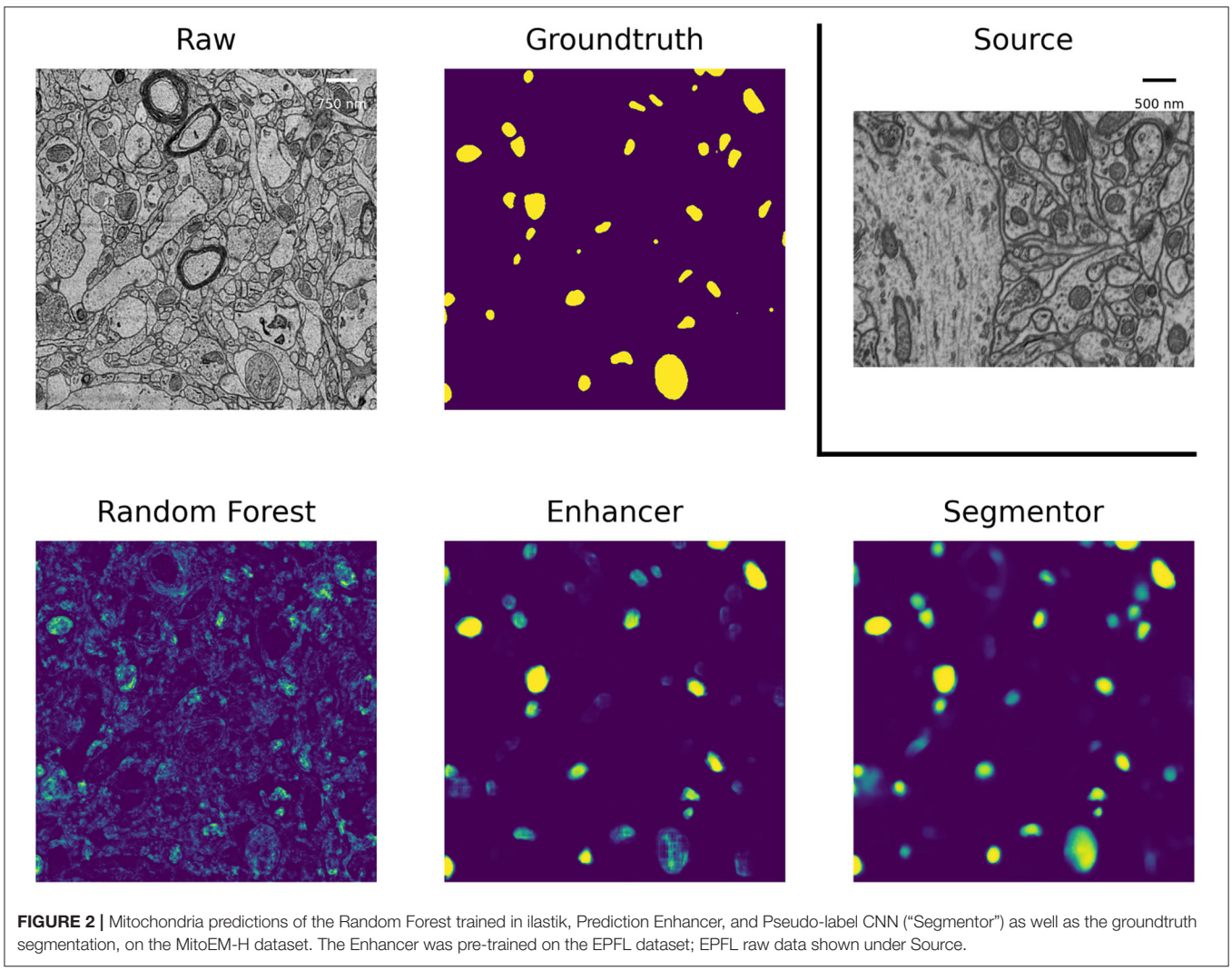


FIGURE 2 | Mitochondria predictions of the Random Forest trained in ilastik, Prediction Enhancer, and Pseudo-label CNN ("Segmentor") as well as the groundtruth segmentation, on the MitoEM-H dataset. The Enhancer was pre-trained on the EPFL dataset; EPFL raw data shown under Source.

Table 3 shows that this is indeed the case, especially for the Kasthuri dataset.

3.3. Membrane Segmentation

We perform membrane segmentation both in EM and LM data. Obtaining a (semantic) membrane segmentation is often the first step in methods for instance segmentation of neurons or cells as direct prediction of an instance segmentation with a CNN is highly non-trivial due to the label invariance problem. As a consequence we are interested in the quality of the final instance segmentation, not the intermediate boundary segmentation, in these experiments and set up a up a Multicut based post-processing procedure similar to Beier et al. (2017) to obtain instances from the boundary predictions. We then evaluate the instance segmentation using the Variation of Information (Meilă, 2003). Direct evaluation of the boundary predictions via the F1-score is often not indicative of the quality of the resulting instance segmentation due to the large influence of relatively small prediction errors, such as holes (Arganda-Carreras et al., 2015). For the Variation of Information lower values correspond to a better segmentation.

In EM we perform boundary segmentation of neural tissue using the VNC dataset as source and three different datasets from the CREMI challenge (cremi.org) as target. **Table 4** shows that the PE significantly improves the RF predictions for all three target datasets. The network trained on pseudo-labels can further improve results, especially for CREMI B and C, which pose a more challenging segmentation problem due to more irregular and elongated neurites compared to CREMI A. Both PE and Pseudo-label Net perform significantly better than a segmentation network trained on the source dataset. The segmentation results of a segmentation network trained on a separate split of the target dataset are shown to indicate an upper bound of the segmentation performance. **Figure 3** shows the improvement brought by the PE and the Pseudo-label Net on an image from CREMI C.

In LM we perform boundary segmentation of cells in a confocal microscopy image stack of *Arabidopsis thaliana* ovule tissue. We use a light-sheet microscopy image stack of Arabidopsis root tissue as source data. Note that we downsample the Ovules dataset by a factor of 2 to match the resolution of the Root dataset (see **Table 1B**). The results are shown in the "Root (LM)" column in **Table 5**. The PE significantly improves the RF results and pseudo-label training improves them even further. In this case the quality of the pseudo-label net almost reaches the target network. Note that the overall quality of results reported here is inferior compared to the results reported in Wolny et al. (2020). This can be explained by the fact that all models only receive 2D input, whereas the state-of-the-art uses 3D models.

We also experiment with a much larger domain shift and apply a PE that was trained on the EM dataset CREMI A as source. The results are shown in the "CREMI (EM)" column in **Table 5**. As expected, transfer of the source network fails, because it was trained on a completely different domain. However, the PE successfully improves RF predictions. The fact that the PE only receives the RF predictions as input enables successful transfer in this case; while the image data distribution is very different in source and target domain, RF probability maps look sufficiently similar. Furthermore, the resolution of the two domains differs by almost three orders of magnitude. However, the size of the structures in pixels is fairly similar, enabling successful domain adaptation. **Figure 4** shows RF, PE and Pseudo-label Net predictions next to the source and target domain data. In this case, training with pseudo-labels does not improve the result, probably because the predictions get smoothed significantly compared to the PE, as can be seen in the figure.

TABLE 3 | Mitochondria segmentation results for PE trained on multiple source datasets.

Source	EPFL	VNC	MitoEM-R	MitoEM-H	Kasthuri
EPFL	0.811	0.786	0.627	0.505	0.612
EPFL, VNC	0.806	0.818	0.642	0.515	0.672
EPFL, VNC MitoEM-R, MitoEM-H	0.833	0.832	0.675	0.586	0.720

The left column indicates the source datasets, quality is measured with the F1 score.

TABLE 4 | Results for boundary segmentation in EM.

Model/Dataset	CREMI A	CREMI B	CREMI C
Source net	1.031	2.089	1.925
RF	1.092	2.231	1.797
PE	0.856	2.107	1.756
Pseudo-label net	**0.840**	**1.806**	**1.593**
Target net	0.559	0.739	1.055

Quality is measured by the Variation of Information (lower is better) after instance segmentation via Multicut post-processing. Source Net and PE are trained on the VNC dataset and then applied to the three target datasets CREMI A, B, and C. RF is trained interactively with ilastik on each target dataset. Best result is shown in bold.

3.4. Nuclei Segmentation

As another example of cross-modality adaptation, we perform nucleus segmentation between fluorescence microscopy images from Caicedo et al. (2019) (DSB-FL) and histopathology images of the human kidney from Kumar et al. (2019) (Monuseg). **Table 6** shows the results for using Monuseg as source and DSB-FL as target (column "DSB-FL") and vice versa (column "Monuseg"). The Enhancer and pseudo-label training offer a modest improvement for the transfer from Monuseg to DSB-FL. For the transfer in the opposite direction the Enhancer yields inferior results compared to ilastik predictions and consequently also inferior results for pseudo-label training. This observation can be explained by the fact that the images in the DSB-FL dataset were acquired with different microscopy modalities and resolutions, resulting in significantly different nuclei sizes across the dataset. In contrast, the size of nuclei in the Monuseg dataset is uniform and closest to the smallest nuclei in DSB-FL. We identify this behavior as a limitation of our method and further investigate the results in **Table 9**.

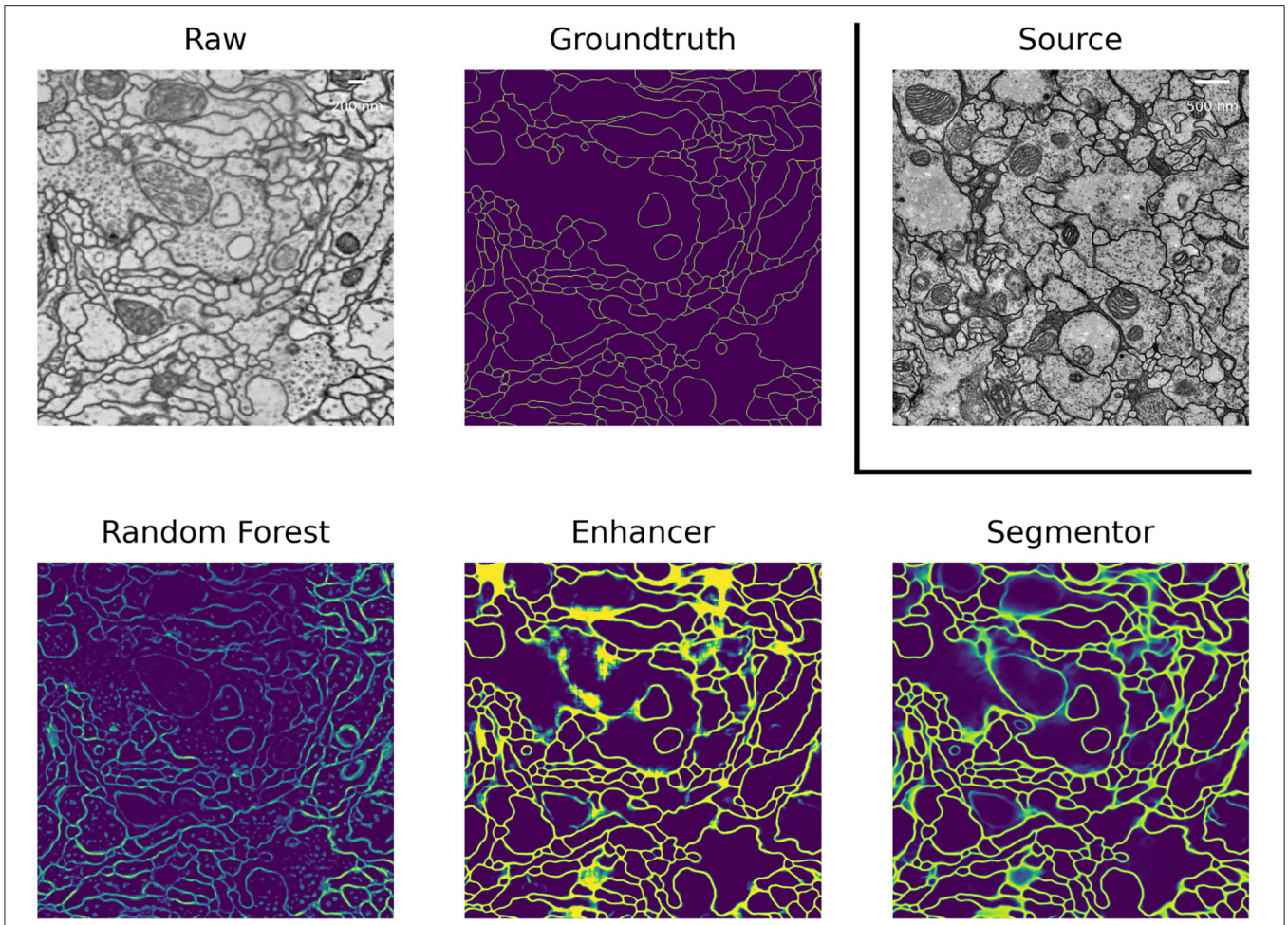


FIGURE 3 | Boundary predictions of the Random Forest trained in ilastik, Prediction Enhancer, and Pseudo-label Net, as well as the groundtruth segmentation, on the CREMI C dataset. The Enhancer was pre-trained on VNC, VNC raw data shown under Source.

TABLE 5 | LM-Boundaries and cross modality experiments: Variation of Information after applying graph partitioning (Multicut) to the boundary predictions.

Model/Source	Root (LM)	CREMI (EM)
Source net	1.782	3.257
RF	1.891	1.891
PE	1.576	**1.605**
Pseudo-label net	**1.563**	1.834
Target net	1.561	1.561

Best result is shown in bold.

3.5. Ablation Studies

In the following, we perform ablation studies to determine the impact of some of our design choices on the overall performance of the method.

First, we investigate if the consistency loss (CL, Equation 6) and label rectification (LR, Equation 1) improve the accuracy obtained after pseudo-label training. We perform pseudo-label training for mitochondria segmentation on the VNC and MitoEM-R datasets using the PE trained on VNC to generate the pseudo-labels. We perform the training without any modification of the loss, adding only CL, adding only LR and adding both CL and LR. The results in **Table 7** show that both CL and LR improve performance on their own. Combining them leads to an additional small improvement on VNC and to a slight decrease in quality on MitoEM-R.

Using the same experiment setup, we also investigate whether using the PE enhancer for generating the pseudo-labels is actually beneficial compared to using the RF trained on target or using the source network. **Table 8** shows that using the PE for pseudo-label generation significantly improves over the two other approaches. We have also studied the influence of the size of the Random Forests used for training the PE, but found that it did not have a significant influence on PE performance. See **Table A2** for details.

3.6. Limitations

The high number of layers, their interconnections and especially skip-connections between them allow the U-net to implicitly

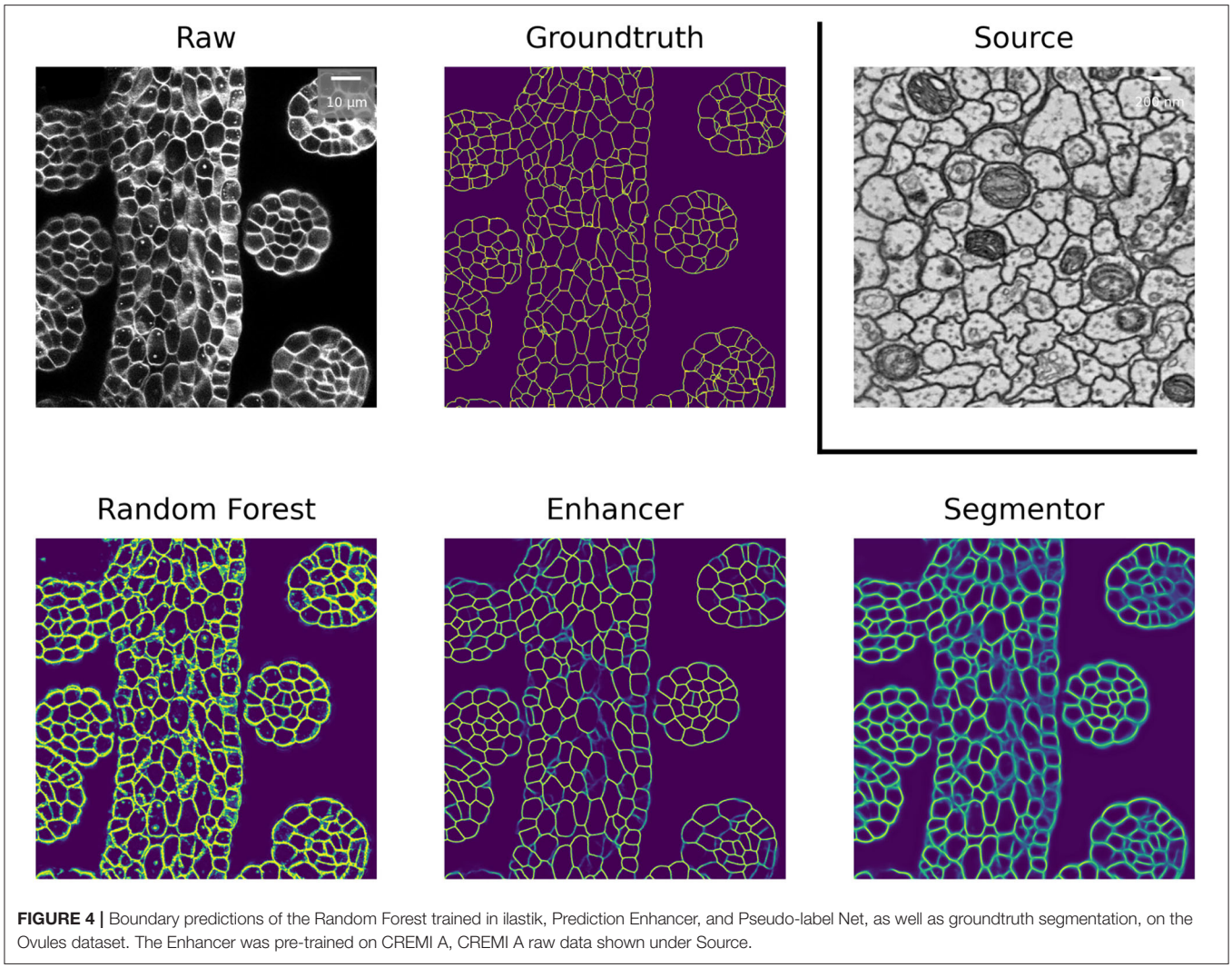


FIGURE 4 | Boundary predictions of the Random Forest trained in ilastik, Prediction Enhancer, and Pseudo-label Net, as well as groundtruth segmentation, on the Ovules dataset. The Enhancer was pre-trained on CREMI A, CREMI A raw data shown under Source.

TABLE 6 | Results of nucleus segmentation.

Source	Method/Target	DSB-FL	Monuseg
	Ilastik	0.856	0.601
DSB-FL	Source net	–	0.014
	Enhancer	–	0.620
	Pseudo-label net	–	**0.654**
Monuseg	Source net	0.001	–
	Enhancer	0.669	–
	Pseudo-label net	0.730	–
	Target net	0.936	0.721

DSB-FL columns shows results for domain adaptation from Monuseg (Histopathology) to DSB-FL (Fluorescence), Monuseg column shows the opposite. The segmentation quality is measured by the F1 score, best result shown in bold.

TABLE 7 | Results of pseudo-label network training using different loss functions.

Method/Dataset	VNC	MitoEM-R
PE	0.840	0.705
Pseudo-labels	0.869	0.768
Pseudo-labels + CL	0.877	0.788
Pseudo-labels + LR	0.869	**0.798**
Pseudo-labels + CL + LR	**0.884**	0.793

Mitochondria segmentation with EPFL as source dataset and VNC, MitoEM-R as target datasets. Segmentation accuracy is measured by the F1 score, best result shown in bold.

learn a strong shape prior for the objects of interest. This effect is exacerbated in our Prediction Enhancer network as it by design does not observe the raw pixel properties and has to exploit shape cues even more than a regular segmentation U-net. While this effect is clearly advantageous for same-task transfer learning, it can lead to catastrophic network hallucinations if very differently shaped objects of interest need to be segmented in the target domain. To illustrate this point, we show the transfer of a PE learned for mitochondria on the EPFL dataset to predict boundaries on the VNC dataset and vice versa in **Figure 5**. The PE amplifies/hallucinates the structures it was trained on while suppressing all other signal in the prediction.

Besides the hallucinations observed in the case of very different shapes of objects in source and target, the size distribution of objects also matters. In Section 3.4, we have investigated transfer between nuclei imaged in histopathology and fluorescence microscopy and observed that the Enhancer yields inferior results for the transfer from histopathology to fluorescence. This can possibly be explained by the fact that the fluorescence dataset contains images of different modalities and resolution, in which the nuclei appear in different sizes. In some of the images the nuclei are small and have a similar average size as in the histopathology dataset, in another one they are of medium size and in yet another of much larger size. We have split the fluorescence dataset into these three modalities ("Small," "Medium," "Large") and list the corresponding results in **Table 9**. The quality of the Enhancer and pseudo-label network predictions drops dramatically for large nuclei sizes, bringing us to hypothesize that such a significant difference in object size constitutes a domain shift our method cannot easily address, even if the underlying problem is so simple it can almost be solved by the Random Forest alone.

A further potential limitation for our method are systematic differences between the error characteristics of the shallow classifiers used for training on the source dataset and the Random Forest used during inference on the target dataset. We set up a synthetic experiment to investigate this case and train the Enhancer using a mixture of the Random Forest predictions and ground-truth labels as input. **Table 10** shows the results

TABLE 8 | Results of pseudo-label network training using RF, Source Network, and PE for label generation.

Pseudo-labels	VNC	MitoEM-R
RF	0.546	0.648
RF w/ CL + LR	0.584	0.656
Source Net	0.707	0.754
Source Net w/ CL + LR	0.794	0.765
PE	0.869	0.768
PE w/ CL + LR	**0.884**	**0.793**

Mitochondria segmentation with EPFL as source dataset and VNC, MitoEM-R as target datasets. Segmentation quality is measured by the F1 score, best result shown in bold.

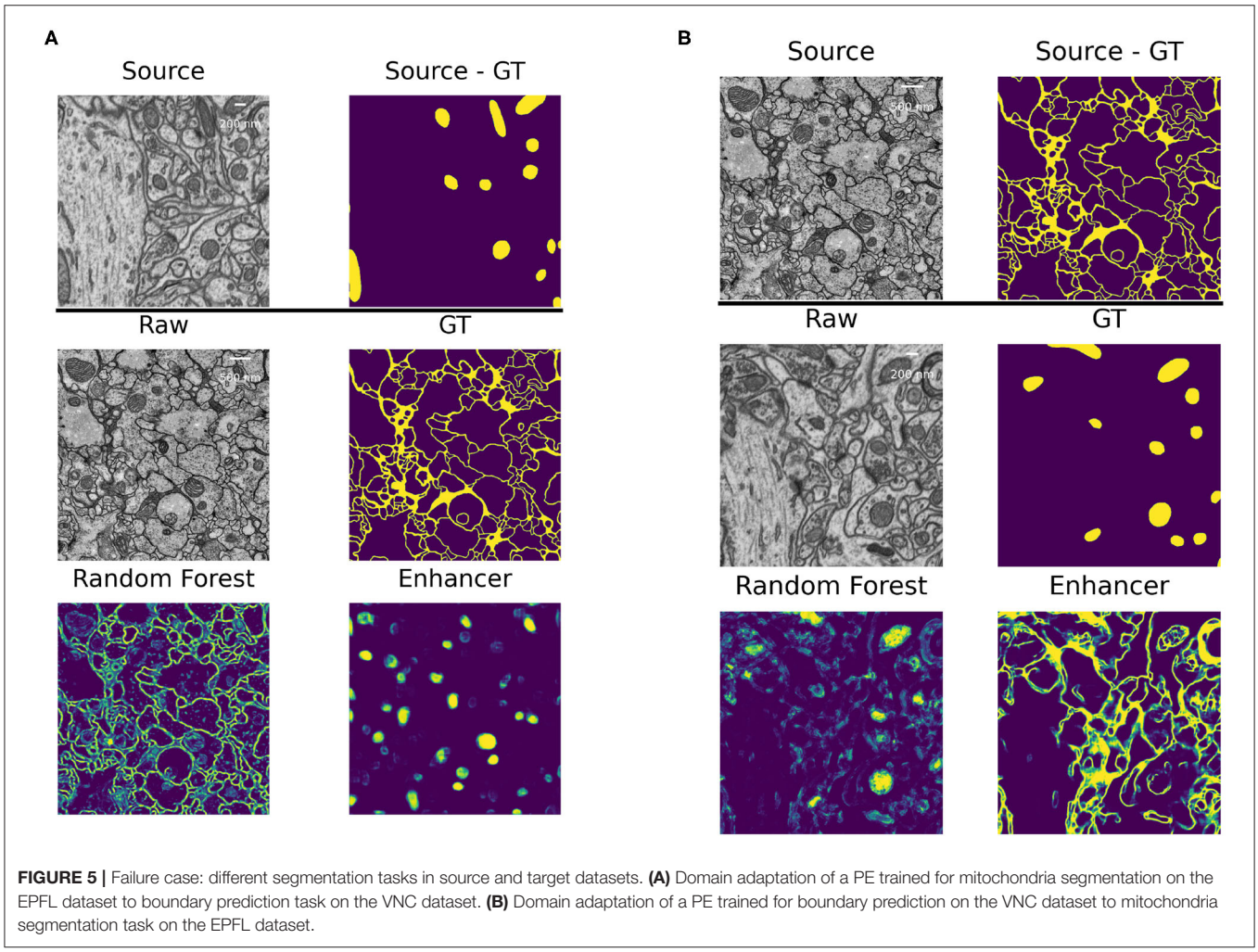


FIGURE 5 | Failure case: different segmentation tasks in source and target datasets. **(A)** Domain adaptation of a PE trained for mitochondria segmentation on the EPFL dataset to boundary prediction task on the VNC dataset. **(B)** Domain adaptation of a PE trained for boundary prediction on the VNC dataset to mitochondria segmentation task on the EPFL dataset.

TABLE 9 | F1-scores for nucleus segmentation in fluorescence microscopy images.

Method/Nucleus size	All	Small	Medium	Large
Ilastik	0.856	0.801	0.851	0.936
Enhancer	0.670	0.784	0.707	0.485
Pseudo-label net	0.730	0.805	0.776	0.592

*We split the dataset into three subsets based on the mean nucleus size per image and obtain 22 images with small nuclei, 12 with medium sized nuclei, and 16 with large nuclei. "All" is referring to the average score for all images and is the same as reported in **Table 6**.*

TABLE 10 | The quality of Enhancer predictions on the target data when trained on a mixture of random-forest and ground-truth label on source.

Mixture/Dataset (Metric)	MitosVNC (F1)	MembranesCremiB (Vol)
RF0%GT100%	0.521	3.113
RF25%GT75%	0.635	2.828
RF50%GT50%	0.697	2.423
RF75%GT25%	0.670	**2.032**
RF100%GT0%	**0.840**	2.107
ilastik	0.647	2.23

For the mitochondria segmentation task we use EPFL as source and VNC as target, the quality is measured by the F1-Score (higher is better). For the membrane segmentation task we use VNC as source and CremiB as target, the quality is measured by the Variation of Information after Multicut segmentation (lower is better). "ilastik" denotes the quality of the Random Forest predictions used on the target, which were obtained by interactive training in ilastik. Best result is shown in bold.

for mitochondria prediction using EPFL as source and VNC as target (cf. Section 3.2) as well as for membrane predictions using VNC as source and CremiB as target (cf. Section 3.3). For both experiments we present the Enhancer network with a weighted linear combination of the smoothed groundtruth and the Random Forest predictions during training and tune the weight coefficient between 0 and 100%. For reference we also report the performance of the ilastik Random Forest that is being "enhanced" on the target dataset. We observe that the prediction quality of the Enhancer is significantly better when trained with a large contribution the Random Forest predictions or from pure Random Forest predictions. We conclude that systematic differences in the errors on source and target, especially if the error rate is significantly lower on source, negatively affect the accuracy of our method.

4. DISCUSSION

We have introduced a simple, source-free, weakly supervised approach to transfer learning in microscopy which can overcome significant domain gaps and does not require adversarial training. In our setup, the feature-based classifier which is trained from sparse annotations on the target domain acts as an implicit domain adapter for the Prediction Enhancer network. The combination of the feature-based classifier and the prediction enhancer substantially outperforms the segmentation CNN trained on the source domain, with further improvement brought by an additional training step where the Enhancer predictions on the target dataset serve as pseudo-labels. Since the Enhancer network never sees the raw data as input, our method can perform transfer learning between domains of drastically different appearance, e.g., between light and electron microscopy images. By design, this kind of domain gap cannot be handled by unsupervised domain adaptation methods which rely on network feature or raw data alignment. Furthermore, even for small domain gaps and in presence of label rectification strategies, pseudo-labels produced by the Prediction Enhancer lead to much better segmentation CNNs than pseudo-labels of the source network. We expect these results to improve even further with the more advanced label rectification approaches which are now actively introduced in the field.

The major limitation of our approach is the dependency on the quality of the feature-based classifier predictions. We expect that in practice users will train it interactively on the target domain, which already produces better results than "bulk" training: in our mitochondria segmentation experiments, also shown in **Table 2**, there was commonly a 1.5- to 2-fold improvement in F1-score between interactive ilastik training in the target domain and RF training in a script without seeing the data. In general, the performance of the Prediction Enhancer will lag behind the performance of a segmentation network trained directly on the raw data with dense groundtruth labels except for very easy problems that can be solved by the RF to 100% accuracy. In a way, the Random Forest acts as a lossy compression algorithm for the raw data, which reduces the discriminative power for the Enhancer. However, the pseudo-label training step can again compensate for the "compression" as it allows to train another network on the raw data of the target domain, with pseudo-labels for potentially very large amounts of unlabeled data.

We have also investigated further limitations of our method and found that it is only applicable if the shape and size distribution of objects in the source and target datasets are sufficiently similar. If this is not the case, the accuracy of our method will drop and, in case of dramatic differences between objects of interest, such as membranes vs. mitochondria, it may even hallucinate structures of similar shape as found in the source data. Furthermore, our method relies on the fact that the data distribution of the Random Forest predictions is closer than the raw data distribution between source and target dataset. Given that we always use the same convolutional filter banks for feature computation, the Random Forests on source and target share the same inductive bias and this assumption will most of the time hold up when segmenting the same semantic class (with similar shape and size distributions). However, in some cases systematic differences between Random Forest predictions on source and target may still exist, for example if the source data has much higher signal-to-noise ratio and thus presents an easier segmentation problem. In this case the segmentation accuracy of our method will suffer despite close shape and size distribution.

For simplicity, and also to sample as many source/target pairs with full groundtruth as possible, we have only demonstrated results on 2D data, in a binary foreground/background classification setting. Extension to 3D is straightforward and

would not require any changes in our method other than accounting for potentially different z resolution between source and target datasets. Extension to multi-class segmentation would only need a simple update to the pseudo-label training loss.

In future work, we envision integration of our approach with other pseudo-label training strategies. Furthermore, as pseudo-label training can largely be configured without target domain knowledge, we expect our method to be a prime candidate for user-facing tools which already include interactive feature-based classifier training.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

AK, AM, AW, and CP have conceptualized the method. AM has implemented the method and run the experiments under the supervision of AK, AW, and CP. AM and CP have drafted the manuscript. AK, AW, and CP have written the final manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

AW was funded by DFG FOR2581 for this work.

ACKNOWLEDGMENTS

We thank the EMBL IT Services for their support.

REFERENCES

Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K. W., Schindelin, J., Cardona, A., et al. (2017). Trainable Weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics* 33, 2424–2426. doi: 10.1093/bioinformatics/btx180

Arganda-Carreras, I., Turaga, S. C., Berger, D. R., Cireşan, D., Giusti, A., Gambardella, L. M., et al. (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. *Front. Neuroanat.* 9:142. doi: 10.3389/fnana.2015.00142

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

Belevich, I., Joensuu, M., Kumar, D., Vihinen, H., and Jokitalo, E. (2016). Microscopy image browser: a platform for segmentation and analysis of multidimensional datasets. *PLoS Biol.* 14:e1002340. doi: 10.1371/journal.pbio.1002340

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., et al. (2019). Ilastik: interactive machine learning for (bio) image analysis. *Nat. Methods* 16, 1226–1232. doi: 10.1038/s41592-019-0582-9

Bermúdez-Chacón, R., Altingövde, O., Becker, C., Salzmann, M., and Fua, P. (2019). Visual correspondences for unsupervised domain adaptation on electron microscopy images. *IEEE Trans. Med. Imaging* 39, 1256–1267. doi: 10.1109/TMI.2019.2946462

Breiman, L. (2001). Random forests. *Mach. Learn.* 45, 5–32. doi: 10.1023/A:1010933404324

Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., et al. (2019). Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nat. Methods* 16, 1247–1253. doi: 10.1038/s41592-019-0612-7

Chen, C., Dou, Q., Chen, H., Qin, J., and Heng, P.-A. (2019). "Synergistic image and feature adaptation: towards cross-modality domain adaptation for medical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33* (Honolulu, HI), 865–872. doi: 10.1609/aaai.v33i01.3301865

Choi, J., Jeong, M., Kim, T., and Kim, C. (2019). Pseudo-labeling curriculum for unsupervised domain adaptation. *arXiv preprint arXiv:1908.00262*.

Du, Y., Yang, H., Chen, M., Jiang, J., Luo, H., and Wang, C. (2021). Generation, augmentation, and alignment: a pseudo-source domain based method for source-free domain adaptation. *arXiv preprint arXiv:2109.04015*.

El Jurdi, R., Petitjean, C., Honeine, P., Cheplygina, V., and Abdallah, F. (2021). High-level prior-based loss functions for medical image segmentation: a survey. *Comput. Vis. Image Understand.* 210:103248. doi: 10.1016/j.cviu.2021.103248

Gerhard, S., Funke, J., Martel, J., Cardona, A., and Fetter, R. (2013). Segmented anisotropic sstem dataset of neural tissue. figshare. Dataset. doi: 10.6084/m9.figshare.856713.v1

Han, S., Pool, J., Narang, S., Mao, H., Tang, S., Elsen, E., et al. (2016). DSD: regularizing deep neural networks with dense-sparse-dense training flow. *arXiv preprint arXiv:1607.04381*.

Januszewski, M., and Jain, V. (2019). Segmentation-enhanced cyclegan. *bioRxiv* 2019:548081. doi: 10.1101/548081

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

Kumar, N., Verma, R., Anand, D., Zhou, Y., Onder, O. F., Tsougenis, E., et al. (2019). A multi-organ nucleus segmentation challenge. *IEEE Trans. Med. Imaging* 39, 1380–1391. doi: 10.1109/TMI.2019.2947628

Liu, D., Zhang, D., Song, Y., Zhang, F., O'Donnell, L., Huang, H., et al. (2020). Pdam: a panoptic-level feature alignment framework for unsupervised domain adaptive instance segmentation in microscopy images. *IEEE Trans. Med. Imaging* 40, 154–165. doi: 10.1109/TMI.2020.3023466

Liu, Y., Zhang, W., and Wang, J. (2021). "Source-free domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Virtual), 1215–1224. doi: 10.1109/CVPR46437.2021.00127

Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning* (Lille), 97-105.

Lucchi, A., Li, Y., and Fua, P. (2013). "Learning for structured prediction using approximate subgradient descent with working sets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Portland, OR), 1987–1994. doi: 10.1109/CVPR.2013.259

Meilă, M. (2003). "Comparing clusterings by the variation of information," in *Learning Theory and Kernel Machines* (Washington, DC: Springer), 173–187. doi: 10.1007/978-3-540-45167-9_14

Prabhu, V., Khare, S., Kartik, D., and Hoffman, J. (2021). S4t: Source-free domain adaptation for semantic segmentation via self-supervised selective self-training. *arXiv preprint arXiv:2107.10140*.

Roels, J., Hennies, J., Saeys, Y., Philips, W., and Kreshuk, A. (2019). "Domain adaptive segmentation in volume electron microscopy imaging," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (Venice), 1519–1522. doi: 10.1109/ISBI.2019.8759383

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Munich: Springer), 234–241. doi: 10.1007/978-3-319-24574-4_28

Rozantsev, A., Salzmann, M., and Fua, P. (2018). Beyond sharing weights for deep domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 801–814. doi: 10.1109/TPAMI.2018.2814042

Taha, A., Shrivastava, A., and Davis, L. (2021). "Knowledge evolution in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual). doi: 10.1109/CVPR46437.2021.01265

Tarvainen, A., and Valpola, H. (2017). "Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results," in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17* (Red Hook, NY; Long Beach, CA: Curran Associates Inc.), 1195–1204.

Wei, D., Lin, Z., Franco-Barranco, D., Wendt, N., Liu, X., Yin, W., et al. (2020). "Mitoem dataset: large-scale 3d mitochondria instance segmentation from EM images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Lima: Springer), 66–76. doi: 10.1007/978-3-030-59722-1_7

Wolny, A., Cerrone, L., Vijayan, A., Tofanelli, R., Barro, A. V., Louveaux, M., et al. (2020). Accurate and versatile 3D segmentation of plant tissues at cellular resolution. *Elife* 9:e57613. doi: 10.7554/eLife.57613

Wu, S., Chen, C., Xiong, Z., Chen, X., and Sun, X. (2021). "Uncertainty-aware label rectification for domain adaptive mitochondria segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer) (Virtual), 191–200. doi: 10.1007/978-3-030-87199-4_18

Xing, F., Bennett, T., and Ghosh, D. (2019). "Adversarial domain adaptation and pseudo-labeling for cross-modality microscopy image quantification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Shenzhen: Springer), 740–749. doi: 10.1007/978-3-030-32239-7_82

Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., and Wen, F. (2021). "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Virtual), 12414–12424. doi: 10.1109/CVPR46437.2021.01223

Zhang, Y., Miao, S., Mansi, T., and Liao, R. (2018). "Task driven generative modeling for unsupervised domain adaptation: application to x-ray image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Granada: Springer), 599–607. doi: 10.1007/978-3-030-00934-2_67

Zhao, Y., Zhong, Z., Luo, Z., Lee, G. H., and Sebe, N. (2021). Source-free open compound domain adaptation in semantic segmentation. *arXiv preprint arXiv:2106.03422*.

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.

APPENDIX
Data and Setup

TABLE A1 | The number of samples used for train, validation and test splits as well as the size of one of the sample in pixels.

Name	EPFL	VNC	MitoEM-R	MitoEM-H	Kasthuri	CREMI
(A) DATA SIZES AND SPLITS USED FOR THE ELECTRON MICROSCOPY DATASETS						
Train samples	165	1	1	1	1	3
Size train samples	165 × 768 × 1,024	12 × 1,024 × 1,024	300 × 4,096 × 4,096	300 × 4,096 × 4,096	75 × 1,613 × 1,463	90 × 1,250 × 1,250
Val samples	1	1	1	1	1	3
Size val samples	40 × 768 × 1,024	4 × 1,024 × 1,024	100 × 4096 × 4,096	100 × 4,096 × 4,096	10 × 1,613 × 1,563	10 × 1,250 × 1,250
Test samples	1	1	1	1	1	3
Size test samples	125 × 768 × 1,024	4 × 1,024 × 1,024	100 × 4,096 × 4,096	100 × 4,096 × 4,096	75 × 1,553 × 1,334	25 × 1,250 × 1,250

Name	Root	Ovules	DSB-FL	Monuseg
(B) DATA SIZES AND SPLITS USED FOR THE LIGHT MICROSCOPY DATASETS				
Train samples	30	42	435	4
Size train samples	355 × 505 × 1,320	340 × 1,035 × 992	325 × 360	1,000 × 1,000
Val samples	2	2	12	1
Size val samples	343 × 535 × 1,165	374 × 1,014 × 1,089	330 × 375	1,000 × 1,000
Test samples	4	6	50	1
Size test samples	373 × 493 × 1,378	373 × 1,200 × 1,094	345 × 390	1,000 × 1,000

Note that we give the averaged sizes in case the size of samples differs across the dataset.

Influence of Number of Random Forests

Here, we study the influence of the number of trees per Random Forest on the Enhancer. We train the Enhancer from RF predictions where each Forest contains 50, 100, 150 or a number of trees drawn randomly from the range 50 to 150. **Table A2** shows the results for the same data as used in Section 3.2 where we have used 100 trees per RF. Note that the results do not directly correspond to any of the results in **Table 2** where we have used further refined target RFs. Here, we observe that the quality of the enhancer is not systematically influenced by the number of trees.

TABLE A2 | F1-scores of the prediction enhancer trained on RF predictions with different numbers of trees for mitochondria segmentation.

	EPFL	VNC	MitoEM-R	MitoEM-H	Kasthuri
50 trees	0.809	0.770	0.607	0.492	0.652
100 trees	0.811	0.786	0.627	0.505	0.612
150 trees	0.811	0.791	0.614	0.504	0.619
50–150 trees	0.814	0.802	0.634	0.525	0.595

EPFL is the source dataset.

frontiers in Computer Science

ORIGINAL RESEARCH
published: 31 March 2022
doi: 10.3389/fcomp.2022.777458

"Tonga": A Novel Toolbox for Straightforward Bioimage Analysis

Alexandra Ritchie[1], *Suvi Laitinen*[1], *Pekka Katajisto*[1,2,3]* *and Johanna I. Englund*[1]*

[1] *Institute of Biotechnology, HiLIFE, University of Helsinki, Helsinki, Finland,* [2] *Department of Cell and Molecular Biology (CMB), Karolinska Institutet, Stockholm, Sweden,* [3] *Faculty of Biological and Environmental Sciences, University of Helsinki, Helsinki, Finland*

Techniques to acquire and analyze biological images are central to life science. However, the workflow downstream of imaging can be complex and involve several tools, leading to creation of very specialized scripts and pipelines that are difficult to reproduce by other users. Although many commercial and open-source software are available, non-expert users are often challenged by a knowledge gap in setting up analysis pipelines and selecting correct tools for extracting data from images. Moreover, a significant share of everyday image analysis requires simple tools, such as precise segmentation, cell counting, and recording of fluorescent intensities. Hence, there is a need for user-friendly platforms for everyday image analysis that do not require extensive prior knowledge on bioimage analysis or coding. We set out to create a bioimage analysis software that has a straightforward interface and covers common analysis tasks such as object segmentation and analysis, in a practical, reproducible, and modular fashion. We envision our software being useful for analysis of cultured cells, histological sections, and high-content data.

Keywords: bioimage analysis, software, segmentation, concave point, intensity measurement, toolbox

OPEN ACCESS

Edited by:
Florian Levet,
UMR5297 Institut Interdisciplinaire de Neurosciences (IINS), France

Reviewed by:
Beth A. Cimini,
Broad Institute, United States
Gaelle Letort,
Institut Pasteur, France

***Correspondence:**
Johanna I. Englund
johanna.englund@helsinki.fi
Pekka Katajisto
pekka.katajisto@helsinki.fi

Specialty section:
This article was submitted to Computer Vision, a section of the journal Frontiers in Computer Science

Received: *15 September 2021*
Accepted: *23 February 2022*
Published: *31 March 2022*

Citation:
Ritchie A, Laitinen S, Katajisto P and Englund JI (2022) "Tonga": A Novel Toolbox for Straightforward Bioimage Analysis. Front. Comput. Sci. 4:777458. doi: 10.3389/fcomp.2022.777458

INTRODUCTION AND RELATED WORK

Analysis of biological processes in quantitative manner through image acquisition has become a widespread technique used across life sciences (Danuser, 2011). Several computational methods have been created to analyze various aspects of biological imaging data, as demonstrated by the numbers of different software available (Levet et al., 2021). Recently, biological questions and bioimaging have become increasingly multifaceted, and can thus lead to generation of vast quantities of data and to development of complex and often custom-made scripts and pipelines (Meijering et al., 2016). There are various efforts and new processing tools created to tackle this issue, including use of artificial intelligence (AI) and deep learning (DL) assisted methods (Litjens et al., 2017; Gomez-de-Mariscal et al., 2021; Hallou et al., 2021; Szkalisity et al., 2021; von Chamier et al., 2021). However, many of these tools require advanced computational expertise or knowledge in bioimage analysis limiting their use. Nevertheless, bioimage analysis tasks routinely performed in life science laboratories require basic tools, including accurate object segmentation, intensity measurements and object classification. Users without in-depth expertise on bioimage analysis or computer sciences are also often challenged by knowledge gap in creating complex analysis pipelines or reproducing custom-made pipelines generated by other users (Carpenter et al., 2012).

Different open-source software including ImageJ, CellProfiler, Icy, QuPath, and Ilastik (Carpenter et al., 2006; de Chaumont et al., 2012; Schindelin et al., 2012; Schneider et al., 2012; Bankhead et al., 2017; Berg et al., 2019) already aim for straightforward analysis of specific bioimage data. ImageJ and CellProfiler are perhaps the most established and widely used image analysis software (Levet et al., 2021). ImageJ, and especially the Fiji distribution of ImageJ, contain various plug-ins and scripts for image visualization, processing, and analysis. Advanced users can readily customize their analysis tools via scripting, and the analyses can be further automated by users recording their own macros (Schneider et al., 2012), which enable replicating the analysis features later. Fiji includes several plug-ins, which are additions to the software and often created for specialized analysis, such as for tracking of objects or image processing (Schindelin et al., 2015). While the available number of tools for ImageJ is large, the choice of appropriate tool and parameters, and how they are combined in pre- and post-processing, can be challenging. CellProfiler, in turn is more modular in its functions, and thus users can easily create specific analysis pipelines (Carpenter et al., 2006). However, we found that whereas efficient and reproducible use of these and other software is uncomplicated for the expert users, there is still a need for user-friendly software with easy access to bioimage analysis for novice users. Thus, we sought to develop a software with an intuitive graphical user-interface, which is easy to operate, and performs the most common bioimage analysis tasks accurately in a reproducible and modular fashion. Special attention was paid to streamlining the combination of segmentation, cell counting, and fluorescent intensities measurement, while providing support for both small and large data sets. We wanted to provide users with automated protocols including all analysis stages from pre-processing to the numerical results, that could seamlessly combine further analysis in a preferred spread sheet program, and designed the software to contain a wizard feature, which aims to suggest a suitable protocol for the analysis to help users select the right tools.

METHODS

Overview

The software is based on Java and presents a graphical user interface. The main software window is divided into the left panel listing all the imported images and their layers, a zoomable main image panel, a separate zoom image panel for viewing image details, and a right tabbed panel, containing method parameters and execution, result panel, and a histogram panel (**Figure 1**). Additional information on the usage of the available filters, protocols, and user interface components and are displayed on the status bar during mouse hover.

The main functionality is achieved through three main components: filters, protocols, and counters. Filters are implementations of general-purpose image processing tools, such as thresholding, gamma and illumination correction, blurring, edge detection, and morphological operations. Filters accept and produce one input and output image. Filters can be used by the user manually, and they are widely utilized internally by the protocols. Protocols are combinations of various subsequently executed filters, additional supplementary code, and a final counter. They provide a full method for processing a series of images for a specific task, such as tissue or nucleus detection and extract numerical data. They accept and can produce one or more input and output images. Counters are special functions for outputting only numerical data from the images into a results table, which can be further saved as TSV or opened in Excel. Full list of available filters and protocols with usage descriptions are available in the user guide (See availability).

Functionality

Various protocols are available via the wizard feature, which aims to suggest a suitable method by asking the user questions about the images and the task the user wants to execute. The protocols and their main settings have been organized into a decision tree, utilized by the wizard (**Figures 2A,B**). Relevant questions from the tree with example images and explanations are presented to the user in a simple "yes or no"-format, and a suitable protocol and settings are selected based on the combination of these choices (**Figures 2B,C**). This eliminates the need to combine filters or search for the suitable methods manually.

Parameters, such as nucleus size, background, radiuses and thresholds required for the protocols are estimated automatically on the go, and the protocol setup requires mainly user interaction related to the input images, and subjective values and preferences such as the desired sensitivity, or removal of dividing cells. Protocols are suitable for both small and large-scale analyses and feature memory mapping and multithreading for efficiently handling a large number of images. There is also a separate batch processing mode for processing several images without importing them before processing.

Most of the common scientific image file formats are supported by Tonga, as the software uses the Bio-Formats importer by Open Microscopy Environment (OME) (Linkert et al., 2010). No knowledge of image formats is required from the user, and the software internally handles imported files as either 32-bit ARGB-images (8 bits per channel), or as 16-bit colored grayscale images, converting them as necessary without user interaction. As the software is currently aimed for 2D images, three-dimensional Z-stack images are automatically converted into average intensity Z-projections. Imported images are displayed to the user as a list (**Figure 1A**; no 2–3), and each image further consist of separate layers, including all fluorescence channels, as well as processed versions of the image produced by the user via filters or protocols. Images can be viewed separately, or as a combined stack image. Any output from the filters or protocols is appended to the list of layers for the image, allowing user to compare the output to the original picture and return to the previous steps.

Protocols

Presently Tonga offers basic protocols for nuclei detection, overlapping and touching nuclei separation, and intensity measurements from fluorescence images, and analysis of tasks such as cell positivity, dye intensity, double-staining intensity,

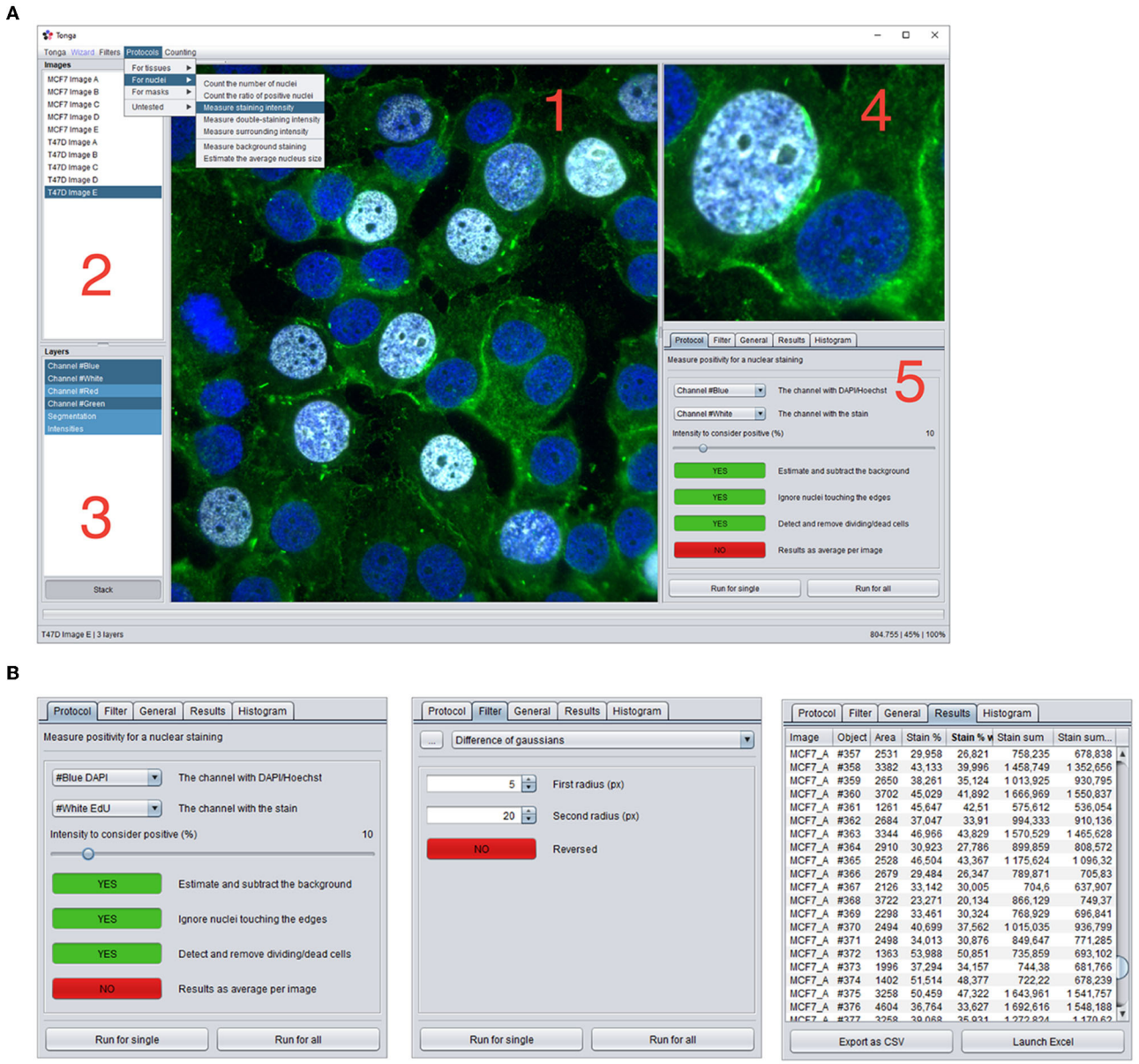


FIGURE 1 | Overview of the Tonga user-interface. **(A)** Main user-interface of Tonga. 1: Main image panel, 2: List of imported images, 3: List of layers (channels and processed versions of images) 4: Zoom panel, 5: Tab panel. Dropdown menus for lauching filters, protocols and counting functions. **(B)** Examples of the content of the Protocols, Filters, and Results Tab panels.

and intensity around the nucleus. Description of the protocol steps and basic principles is presented below. Technical details are available together with the source code, and the usage of the protocols is described in the user guide (See availability).

Nucleus Detection

Tonga proceeds in nucleus detection in three major steps by combining area detection and concave point detection (Zafari et al., 2017) with a concave point pairing algorithm and secondary mask creation. First, a binarized mask is created from the original image using a series of difference of Gaussians (DoG) edge detection with various radiuses, attempting to identify only the area where the target objects are, and remove majority of the background. Second, the edges of these detected areas are traced and any concave points forming sharp corners are detected. The detected concave points are then connected to each other with lines, utilizing information of the angle and direction of the concave point, as well as the location, proximity, parallelity, and sharpness of the other concave points in attempt to determine where the cells most likely overlap. The masks are cut along these connected lines to separate overlapping objects into individual nuclei. Ultimately,

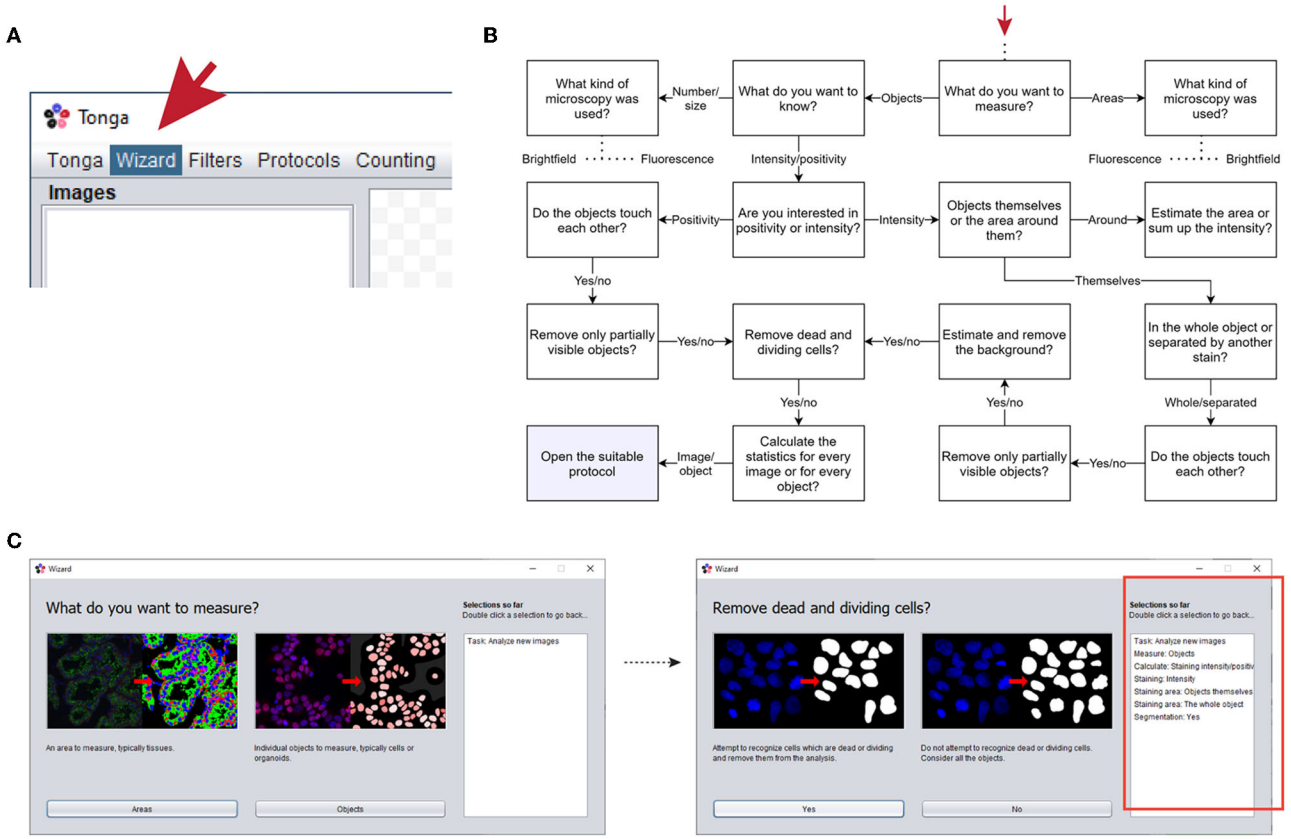


FIGURE 2 | Overview of the wizard feature. **(A)** Wizard can be found and launched from the menu bar. **(B)** Example of one branch of the wizard decision tree, starting point indicated by red arrow. **(C)** Examples of the wizard window helping to select suitable protocol to use. Selections for the analysis made by the user are listed in the box on the right (highlighted with red box in the right image).

the obtained masks are compared back to the original image, in attempt to detect any inconsistencies on the edges, improve mask accuracy, smoothen the mask, and remove unwanted structures, such as additional nuclear structures such as micronuclei and holes caused by nucleoli (Kwon et al., 2020). Dividing and dead cells are removed, per user choice, by detecting nuclei with deviant texture and morphology compared to the other nuclei.

Measurements

The final adjusted mask is used as the nucleus area and compared to the original image for intensity measurements, reported as the average intensity and as the total sum. Background is estimated using an average measurement from one or more areas in the image, which do not produce any signal during the object detection, due to minimal intensity differences. Cell positivity is based on a user-set protocol parameter on the level of intensity required to consider a cell positive.

Modularity

As internally protocols function with an input-output principle by calling existing filters, counters, or other protocols, together with supplementary code, creation of new protocols for various purposes requires only minor code modifications. Implementing new filters also requires minimally only the code which returns an output image array, as the filters share the common internal basic structure, and are executed independently from each other. Indeed, one of the future areas of development in the software is to enable users to extend the software functionality by creating new protocols and loading new filter or counter classes dynamically.

Cell Culture and Immunofluorescence Staining

To generate authentic cell culture data to test our segmentation and intensity measurement data, MCF7, T47D and SKBR-3 breast cancer cell lines (Holliday and Speirs, 2011) (all from ATCC) were cultured on glass coverslips for 24–48 h. Samples were fixed with 4% PFA for 10 min at room temperature. After fixation, cells were washed with PBS and permeabilized with 0.5% Triton-X (Sigma) in PBS for 10 min. Next, samples were washed twice with PBS and non-specific binding sites were blocked with 10% FCS in PBS for 60 min and incubated 120 min at RT with primary antibody (ERα, Santa Cruz 1:100) diluted in blocking solution. Following incubation, samples were washed three times with PBS and then incubated for

60 min at RT with appropriate Alexa Fluor 594 secondary antibody (Life Technologies). Finally, samples were washed for 3 times with PBS followed by counterstaining of nuclei with Hoechst33342 (Sigma). For detection of proliferating cells, cells were treated with 10 μM 5-ethynyl-2′-deoxyuridine (EdU) for 2 h and thereafter fixed with 4 % PFA. Immunostaining to detect EdU positive cells was performed using Click-IT EdU Alexa Fluor 647 Imaging Kit (ThermoFisher) according to manufacturer's instructions. Before mounting, nuclei were counterstained with Hoechst.

Imaging and Quantitation

Imaging of the stained coverslips was performed using Leica DM5000b microscope with 20× objective and 2,048 × 2,048, 0.27 μm/pixel resolution. To test our segmentation method, we prepared MCF7, T47D, and SKBR-3 breast cancer cells stained with Hoechst for counting of nuclei. Two independent human observers counted the nuclei in total of 12 independent images using the Cell Counter function in Fiji. For MCF7 and T47D cells, the nuclei were also classified as either interphase nuclei, or as condensed and small nuclei indicative of mitotic or apoptotic cells.

Comparison With Watershed and StarDist Segmentation in ImageJ

For comparison with watershed segmentation, the images were first pre-processed with illumination correction in Tonga, and then segmented in ImageJ 1.53c by binarizing them with global thresholding and then processing with the watershed operation. For comparison with StarDist segmentation (Schmidt et al., 2018), original RGB images were converted to 8-bit images in ImageJ 1.53c and then processed with the StarDist ImageJ/Fiji plugin (https://imagej.net/StarDist) using the built-in versatile (fluorescent nuclei) model and default settings for post-processing. Processed images were transferred back to Tonga, and any objects touching the images edges or being smaller than 500 pixels (MCF7), 300 pixels (T47D), or 100 pixels (SKBR-3) were removed. The number of segmented areas was counted using the object counter tool in Tonga. Finally, the number of dead and dividing cells estimated by Tonga was subtracted from the total number of objects, to get the number of interphase nuclei. Bland-Altman plots of the results was calculated and drawn in Prism 9.

RESULTS

To demonstrate the usage and accuracy of the automated protocols available in Tonga, we applied segmentation, nuclear counting, stain positivity, and staining intensity protocols to various data sets, and compared the results visually or either to the ground truth numbers or other available methods outside Tonga.

The nuclear segmentation protocol in Tonga was demonstrated by analyzing a nuclear count of a set of MCF7, T47D, and SKBR-3 breast cancer cells (Holliday and Speirs, 2011) by Tonga and two independent human observers (**Figure 3A**). The nuclear counts obtained with Tonga differed 0.1–8% from the observer average, showing good consistency across the image set (**Supplementary Figures 1A,B**). We additionally demonstrated the segmentation protocol in Tonga using fluorescence images from a publicly available, ground truth annotated data set prepared for testing nuclear segmentation methods (Kromp et al., 2020). The data set includes 2D fluorescence images of cell lines and cultured primary cells with varying level of overlapping cells. We segmented 29 images with normal nuclei from the collection using Tonga and compared the number of nuclei against the ground truth annotations of the data sets (**Figure 3B**). For a fair comparison, we used the same criteria for nuclei to be included as in the original work (Kromp et al., 2020), and excluded any nuclei excluded in the ground truth annotations, as well as nuclei which appeared dead or dividing. We found that difference between the ground truth and Tonga was between 0 and 6% across the data set, demonstrating suitability of the nuclear segmentation and counting protocols on various data sets and cell types.

To compare the Tonga segmentation protocol with other existing methods, we analyzed our own data set using widely used watershed-based segmentation with binary thresholding, as well as with state-of-the-art StarDist segmentation method (Schmidt et al., 2018; **Supplementary Figures 1A,B**). For unbiased comparison of only clustered cells, we removed cells with condensed nuclei and micronuclei from the images. We detected no systemic differences or bias in nuclear count with Tonga compared to watershed method or to StarDist when comparing the difference between the methods using a Bland-Altman plot (**Supplementary Figure 1C**). We additionally segmented the Kromp et al. (2020), data set with both Tonga and StarDist to further compare these methods. This data set was also segmented in a similar fashion by Tonga and StarDist, especially when examining clustered nuclei (**Supplemental Figures 1D,E**). Thus, we conclude that the Tonga segmentation protocol functions comparably to both earlier and state-of-the art methods.

Detection of apoptotic and dividing cells largely depends on size and morphology of the nuclei that can vary between different cell types, yet condensed nuclei are indicative of both processes. Condensed nuclei can be removed from or included in the segmentation and counting as per user choice. We demonstrated using the MCF7 cell line that condensed nuclei can be accurately recognized in Tonga using an automated protocol (**Figure 3C**). The final numeric output from the segmentation and counting protocols is displayed to the user in the Results table (**Figure 3D**).

Next, we demonstrated intensity measurements in nuclei in conjunction to nuclear segmentation of nuclei. Measuring of immunofluorescence staining intensity can be used for instance to quantitate number of positive or negative cells in a population, and thus we analyzed MCF7 cells treated with EdU as indication of proliferation (**Figure 4A**). Images were first segmented using the Hoechst nuclear stain channel, whereafter an intensity threshold on channel with background corrected EdU staining was set manually to determine, which nuclei were positive. Cells exhibiting staining intensity below the selected threshold were determined negative, and above the threshold positive (**Figure 4A**, marked in gray and white,

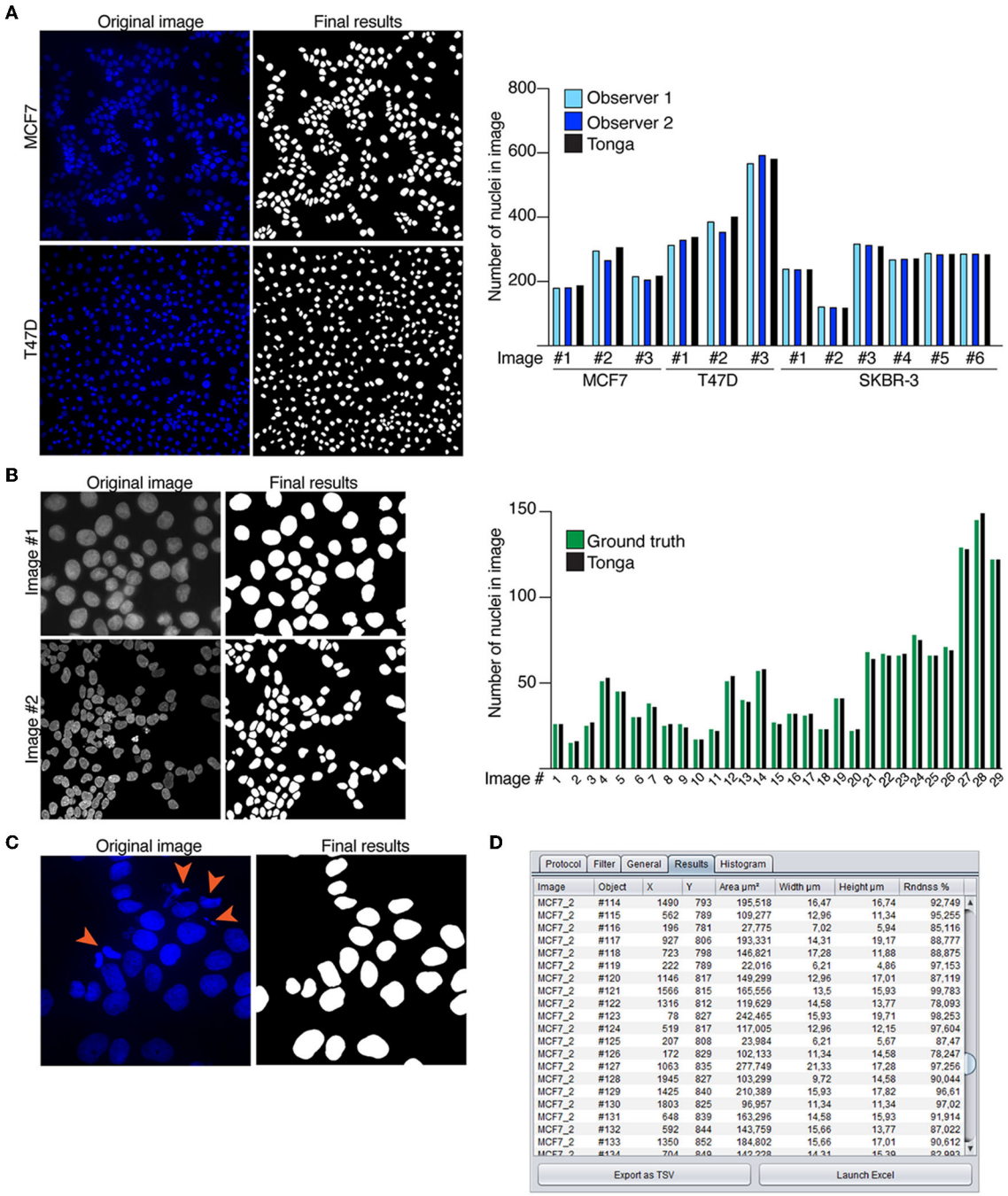


FIGURE 3 | Nuclear segmentation protocols. **(A)** Example images of MCF7 and T47D breast cancer cells with nuclear stain (Hoechst) and corresponding segmentation masks by Tonga. Graph shows quantitation of round nuclei in 12 independent images. Counting was performed manually by two observers, and compared to count by Tonga. **(B)** Example images of Kromp et al. (2020) data set and the final segmentation results created in Tonga. Graph shows nuclear count of 29 images by Tonga compared to the ground truth. **(C)** Example image of MCF7 with nuclear stain (Hoechst). Dividing and apoptotic cells indicated by orange arrowheads in the original image are removed from the final results. **(D)** Numeric output from the nuclear segmentation protocol shown in the results tab.

respectively). When comparing the segmentation mask and original image, it is clear the precision of the original segmentation mask is critical for accurate determination of positive and negative cells (**Figure 4A**). Furthermore, the masks generated by the segmentation protocol appear in the list of layers in the software, which makes it straightforward for the users to examine the data and compare it to the original images.

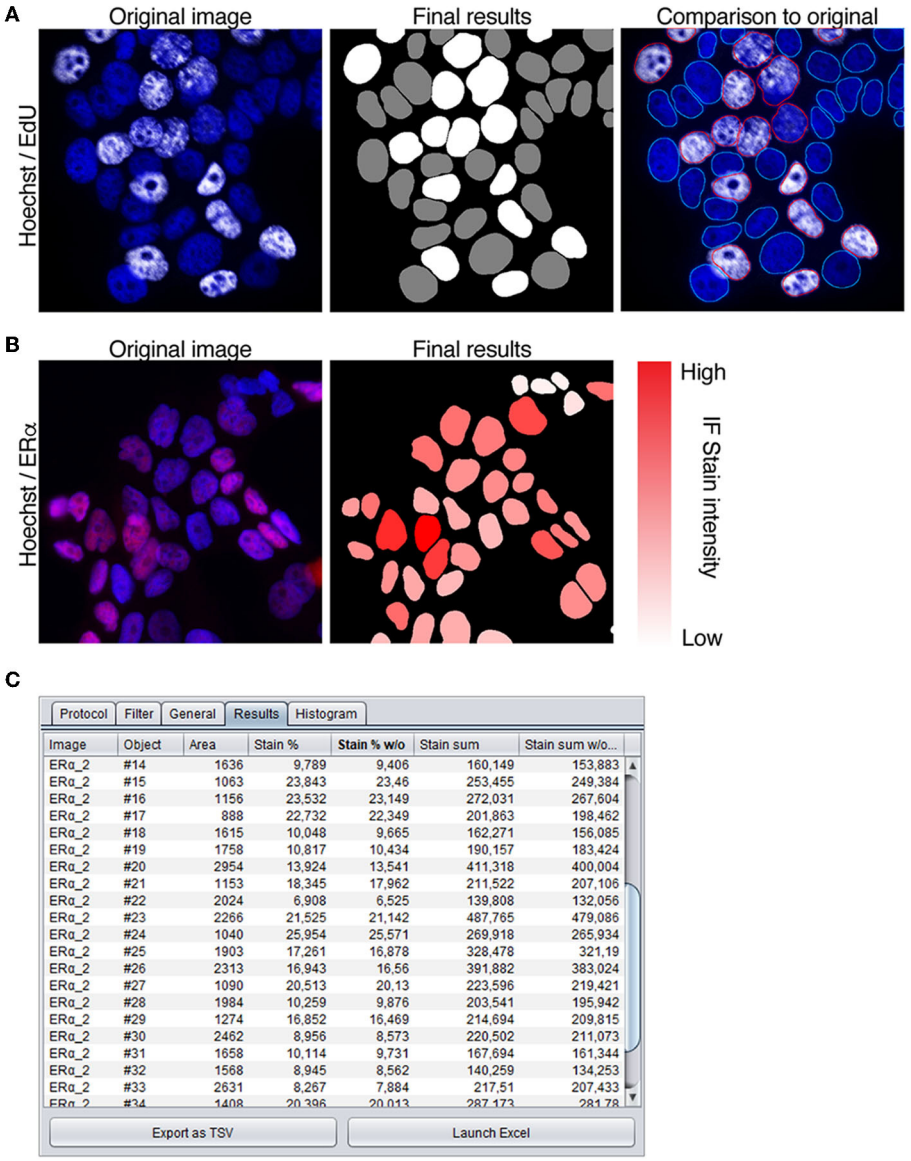


Image	Object	Area	Stain %	Stain % w/o	Stain sum	Stain sum w/o...
ERα_2	#14	1636	9,789	9,406	160,149	153,883
ERα_2	#15	1063	23,843	23,46	253,455	249,384
ERα_2	#16	1156	23,532	23,149	272,031	267,604
ERα_2	#17	888	22,732	22,349	201,863	198,462
ERα_2	#18	1615	10,048	9,665	162,271	156,085
ERα_2	#19	1758	10,817	10,434	190,157	183,424
ERα_2	#20	2954	13,924	13,541	411,318	400,004
ERα_2	#21	1153	18,345	17,962	211,522	207,106
ERα_2	#22	2024	6,908	6,525	139,808	132,056
ERα_2	#23	2266	21,525	21,142	487,765	479,086
ERα_2	#24	1040	25,954	25,571	269,918	265,934
ERα_2	#25	1903	17,261	16,878	328,478	321,19
ERα_2	#26	2313	16,943	16,56	391,882	383,024
ERα_2	#27	1090	20,513	20,13	223,596	219,421
ERα_2	#28	1984	10,259	9,876	203,541	195,942
ERα_2	#29	1274	16,852	16,469	214,694	209,815
ERα_2	#30	2462	8,956	8,573	220,502	211,073
ERα_2	#31	1658	10,114	9,731	167,694	161,344
ERα_2	#32	1568	8,945	8,562	140,259	134,253
ERα_2	#33	2631	8,267	7,884	217,51	207,433
ERα_2	#34	1408	20,396	20,013	287,173	281,78

FIGURE 4 | Intensity measurement protocols. **(A)** MCF7 cells stained with nuclear stain (Hoechst) and to detect EdU (white) incorporated cells. To quantitate EdU positive cells, nuclei are segmented using the Hoechst channel and a threshold is set to determine positive staining. In final results image nuclei that are negative for EdU staining are marked gray, and positive nuclei white. Comparison to original image shows overlay of segmentation mask marked with positive (red) and negative (blue) nuclei **(B)** MCF7 cells immunostained with ERα antibody (red), segmented using nuclear stain channel (Hoechst) and measured for intensity of the staining. Immunofluorescence intensity is indicated with the color slider (white–low intensity, red–high intensity). **(C)** Numeric output of the intensity measurement protocol shown in the results tab.

Combination of segmentation and measuring of immunofluorescence staining intensity can also be used to quantitate the full dynamic range of a nuclear stain (**Figure 4B**). To establish this with our method, MCF7 cells, known to exhibit positive staining (Comsa et al., 2015), were immunostained with antibody against estrogen receptor a (ERα). From the images, nuclei were segmented using the Hoechst channel, and intensity measured from the channel with ERα staining and taking into consideration both the stain intensity and the area of the nucleus. The final mask shows the segmented nuclei overlaid with color ranging from white to red, indicative of the immunostaining intensity (white—low, red—high; **Figure 4B**), and the final numeric results are displayed in the results-table (**Figure 4C**). Our results show that the color corresponds accurately with the brightness of the staining in the original image. These examples demonstrate the suitability of the automated protocols in Tonga for recording immunostaining intensity within nuclei either for quantitative or qualitative analysis.

DISCUSSION

Being user-friendly, modular, and validated are among the key components for a bioimage analysis program to be widely usable for the bioscience community (Carpenter et al., 2012; Levet et al., 2021). Here we describe a validated open-source image analysis software Tonga, which we have developed with the purpose of being practical for everyday image analysis tasks. We paid special attention to the user interface being intuitive, thus making the software straightforward to use and accessible also for users without in-depth knowledge of image analysis. We additionally aimed to develop the software modular and automatic, that combining different analysis tasks would be uncomplicated and analyses would require minimal set up. Therefore, we added the Wizard feature, which helps users to choose the right protocol for their analyses.

We acknowledge there are several excellent open-source software available for variety of complex image analysis tasks (Levet et al., 2021). When developing our software, we especially concentrated on constructing a toolbox for everyday image analysis needs of users that often include segmentation and recording intensity measurements from immunofluorescence images. We show that the protocols we have developed for these analysis work equally well compared to for instance watershed -based algorithms and to deep-learning based methods such as StarDist (Schmidt et al., 2018), especially when nuclei are located close to each other or in aggregates. Moreover, we show our protocols can be effectively used to detect dead and dividing cells and used together with measurement of immunofluorescence staining intensity.

In the future we aim to expand the toolbox with other protocols, including detection and segmentation of tissues and measurement of fluorescence and histochemical stain intensity from histological slides, as well as object detection and segmentation from phase contrast images. The latter would provide analysis tools for example for organoid cultures, and for smaller particles such as organelles within the cell. Furthermore, we are planning to enable a feature in Tonga, where users can integrate their own filters and protocols as a part of the software, to better customize pipelines for their specific needs. As Tonga can analyze several images at a given time, we envision it will be advantageous for large scale image analyses, but also for quick analysis of smaller scale set-up experiments. We believe that the existing features, as well as the future modifications, enable users of different levels operate the system effectively leading to improved access to image analyses and data reproducibility.

DATA AVAILABILITY STATEMENT

A compiled software packages for both Windows and MacOS, as well as the full Java source code of Tonga and all the protocols described in this manuscript are available in GitHub (https://github.com/avritchie/tonga). The user guide for the software is also available in GitHub (https://github.com/avritchie/tonga/wiki). Java version 9 or later is required to run Tonga. The examples shown in this manuscript were made with the development version of the software (0.1.2).

AUTHOR CONTRIBUTIONS

AR, PK, and JE conceptualized and designed the study. AR did primary code development. AR, SL, and JE contributed to the experiments in the manuscript. All authors contributed to the text. All authors contributed to the article and approved the submitted version.

FUNDING

We acknowledge the following sources: Academy of Finland (266869, 304591, and 320185), the Swedish Research Council 2018-03078, Cancerfonden 190634, the Jane and Aatos Erkko Foundation and the Cancer Foundation Finland (PK), and the Finish Cultural Foundation (JE).

ACKNOWLEDGMENTS

The authors would like to thank all the members of the Katajisto laboratory for discussions and feedback regarding the manuscript. Imaging was performed at the Light Microscopy Unit, Institute of Biotechnology, supported by HiLIFE and Biocenter Finland.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2022.777458/full#supplementary-material

REFERENCES

Bankhead, P., Loughrey, M. B., Fernandez, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., et al. (2017). QuPath: Open source software for digital pathology image analysis. *Sci. Rep*. 7, 16878. doi: 10.1038/s41598-017-17204-5

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., et al. (2019). ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* 16, 1226–1232. doi: 10.1038/s41592-019-0582-9

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., et al. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* 7, R100. doi: 10.1186/gb-2006-7-10-r100

Carpenter, A. E., Kamentsky, L., and Eliceiri, K. W. (2012). A call for bioimaging software usability. *Nat. Methods* 9, 666–670. doi: 10.1038/nmeth.2073

Comsa, S., Cimpean, A. M., and Raica, M. (2015). The story of MCF-7 breast cancer cell line: 40 years of experience in research. *Anticancer Res.* 35, 3147–3154.

Danuser, G. (2011). Computer vision in cell biology. *Cell* 147, 973–978. doi: 10.1016/j.cell.2011.11.001

de Chaumont, F., Dallongeville, S., Chenouard, N., Herve, N., Pop, S., Provoost, T., et al. (2012). Icy: an open bioimage informatics platform for extended reproducible research. *Nat. Methods* 9, 690–696. doi: 10.1038/nmeth.2075

Gomez-de-Mariscal, E., Garcia-Lopez-de-Haro, C., Ouyang, W., Donati, L., Lundberg, E., Unser, M., et al. (2021). DeepImageJ: a user-friendly environment

to run deep learning models in ImageJ. *Nat. Methods* 18, 1192–1195. doi: 10.1038/s41592-021-01262-9

Hallou, A., Yevick, H. G., Dumitrascu, B., and Uhlmann, V. (2021). Deep learning for bioimage analysis in developmental biology. *Development* 148, dev199616. doi: 10.1242/dev.199616

Holliday, D. L., and Speirs, V. (2011). Choosing the right cell line for breast cancer research. *Breast Cancer Res.* 13, 215. doi: 10.1186/bcr2889

Kromp, F., Bozsaky, E., Rifatbegovic, F., Fischer, L., Ambros, M., Berneder, M., et al. (2020). An annotated fluorescence image dataset for training nuclear segmentation methods. *Sci. Data* 7, 262. doi: 10.1038/s41597-020-00608-w

Kwon, M., Leibowitz, M. L., and Lee, J. H. (2020). Small but mighty: the causes and consequences of micronucleus rupture. *Exp. Mol. Med*. 52, 1777–1786. doi: 10.1038/s12276-020-00529-z

Levet, F., Carpenter, A. E., Eliceiri, K. W., Kreshuk, A., Bankhead, P., and Haase, R. (2021). Developing open-source software for bioimage analysis: opportunities and challenges. *F1000Res* 10, 302. doi: 10.12688/f1000research.52531.1

Linkert, M., Rueden, C. T., Allan, C., Burel, J. M., Moore, W., Patterson, A., et al. (2010). Metadata matters: access to image data in the real world. *J. Cell Biol.* 189, 777–782. doi: 10.1083/jcb.201004104

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., et al. (2017). A survey on deep learning in medical image analysis. *Med. Image Anal.* 42, 60–88. doi: 10.1016/j.media.2017.07.005

Meijering, E., Carpenter, A. E., Peng, H., Hamprecht, F. A., and Olivo-Marin, J. C. (2016). Imagining the future of bioimage analysis. *Nat. Biotechnol.* 34, 1250–1255. doi: 10.1038/nbt.3722

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Methods* 9, 676–682. doi: 10.1038/nmeth.2019

Schindelin, J., Rueden, C. T., Hiner, M. C., and Eliceiri, K. W. (2015). The ImageJ ecosystem: An open platform for biomedical image analysis. *Mol. Reprod. Dev.* 82, 518–529. doi: 10.1002/mrd.22489

Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. (2018). "Cell detection with star-convex polygons," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Granada, 265–273. doi: 10.1007/978-3-030-00934-2_30

Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). NIH image to ImageJ: 25 years of image analysis. *Nat. Methods* 9, 671–675. doi: 10.1038/nmeth.2089

Szkalisity, A., Piccinini, F., Beleon, A., Balassa, T., Varga, I. G., Migh, E., et al. (2021). Regression plane concept for analysing continuous cellular processes with machine learning. *Nat. Commun.* 12, 2532. doi: 10.1038/s41467-021-22866-x

von Chamier, L., Laine, R. F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., et al. (2021). Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* 12, 2276. doi: 10.1038/s41467-021-22518-0

Zafari, S., Eerola, T., Sampo, J., Kälviäinen, H., and Haario, H. (2017). "Comparison of concave point detection methods for overlapping convex objects segmentation," in *Scandinavian Conference on Image Analysis SCIA 2017: Image Analysis* (Tromsø), 245–256. doi: 10.1007/978-3-319-59129-2_21

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