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.

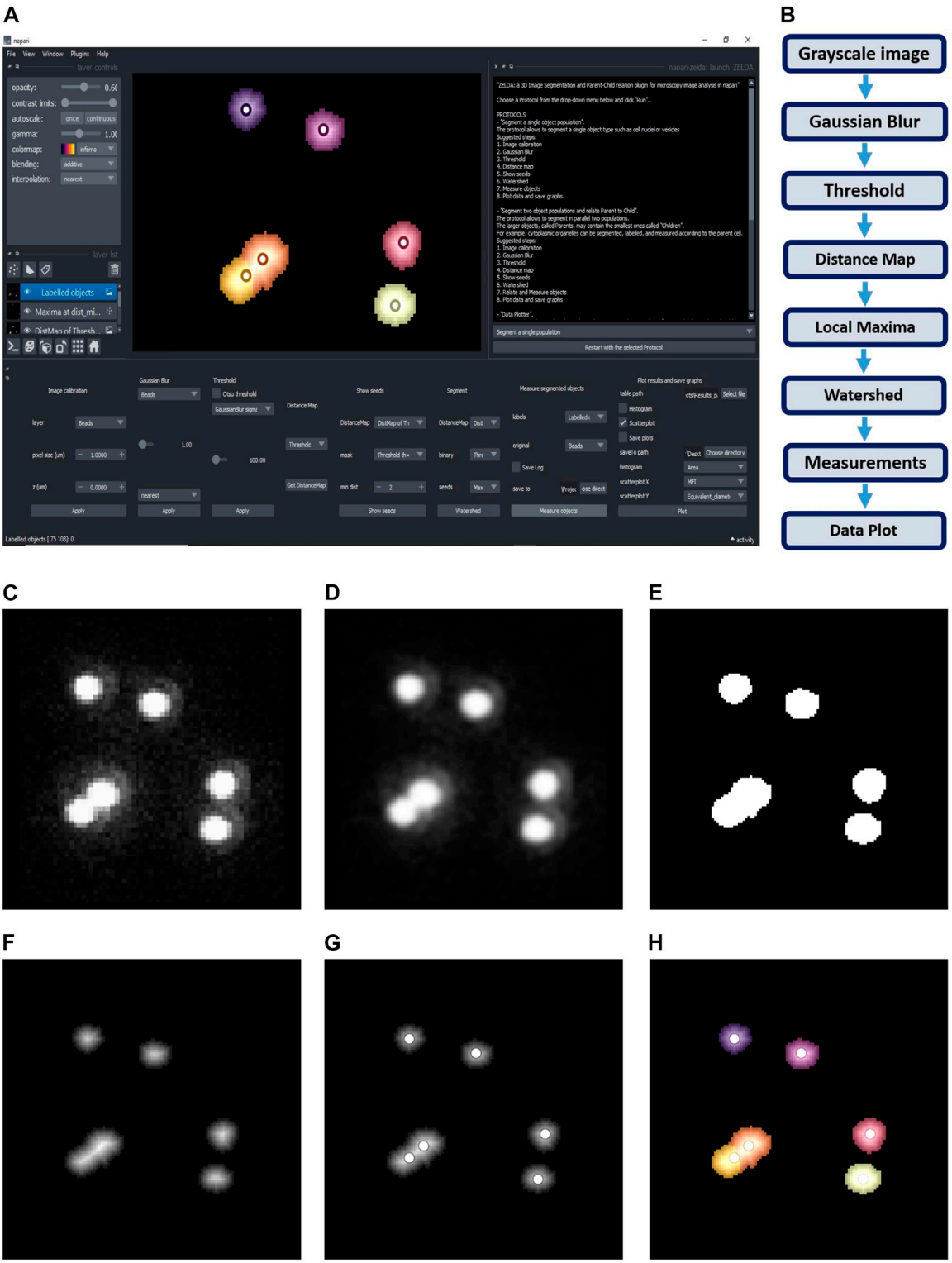

FIGURE 1 | ZELDA plugin for *napari*. **(A)** GUI of ZELDA with the description of the ready-to-use protocols sufficient to run a complete image analysis workflow (white text box on the top right). Each protocol is divided into numbered steps corresponding to the software commands in the *napari* dock 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.

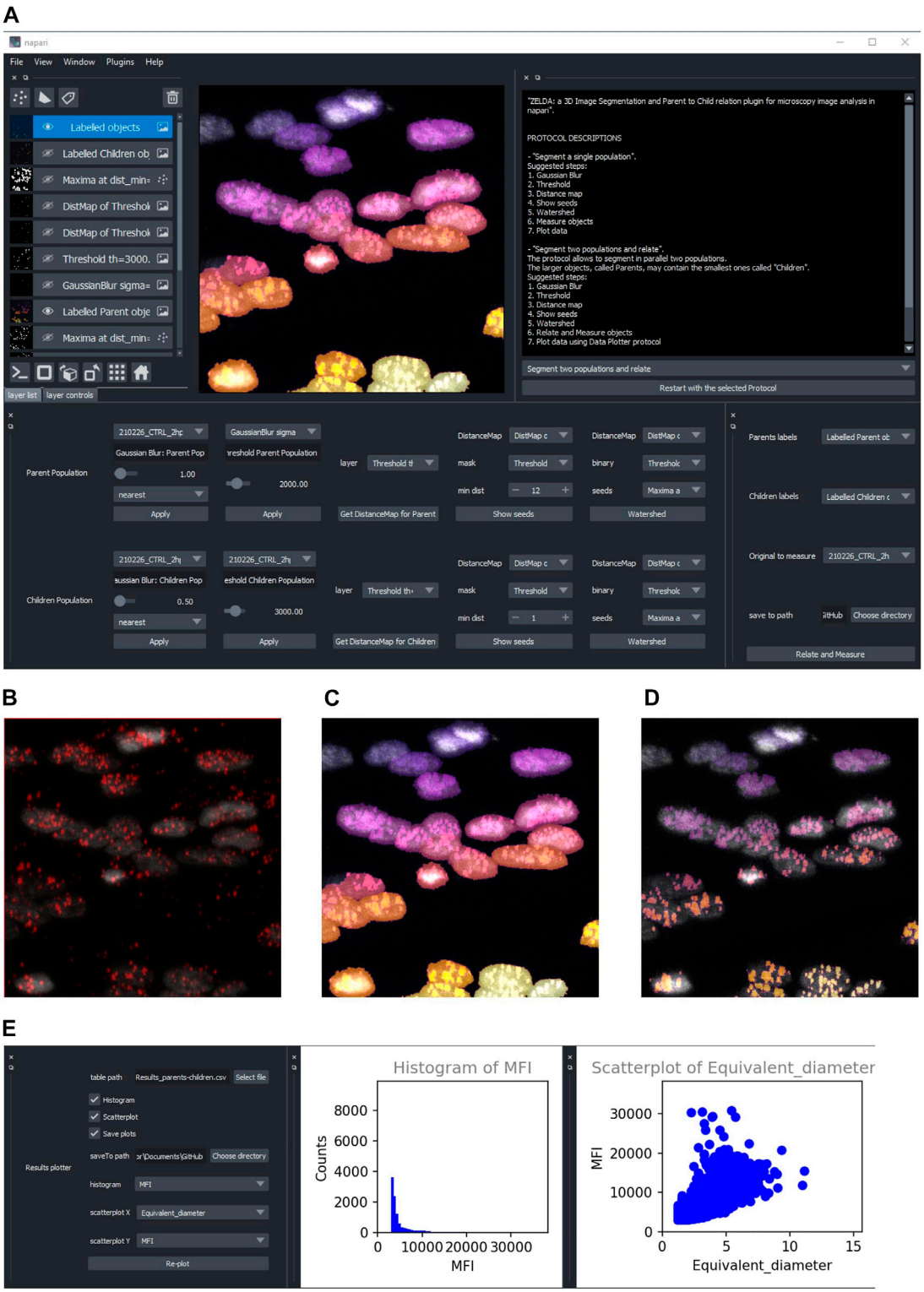

FIGURE 2 | ZELDA application for the 3D segmentation of two object populations and "Parent-child" relation. **(A)** ZELDA protocol "Segment two populations and relate" used to analyze the distribution of viral RNA in infected human cell nuclei. **(B)** Original 3D data set showing a nuclear staining 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.

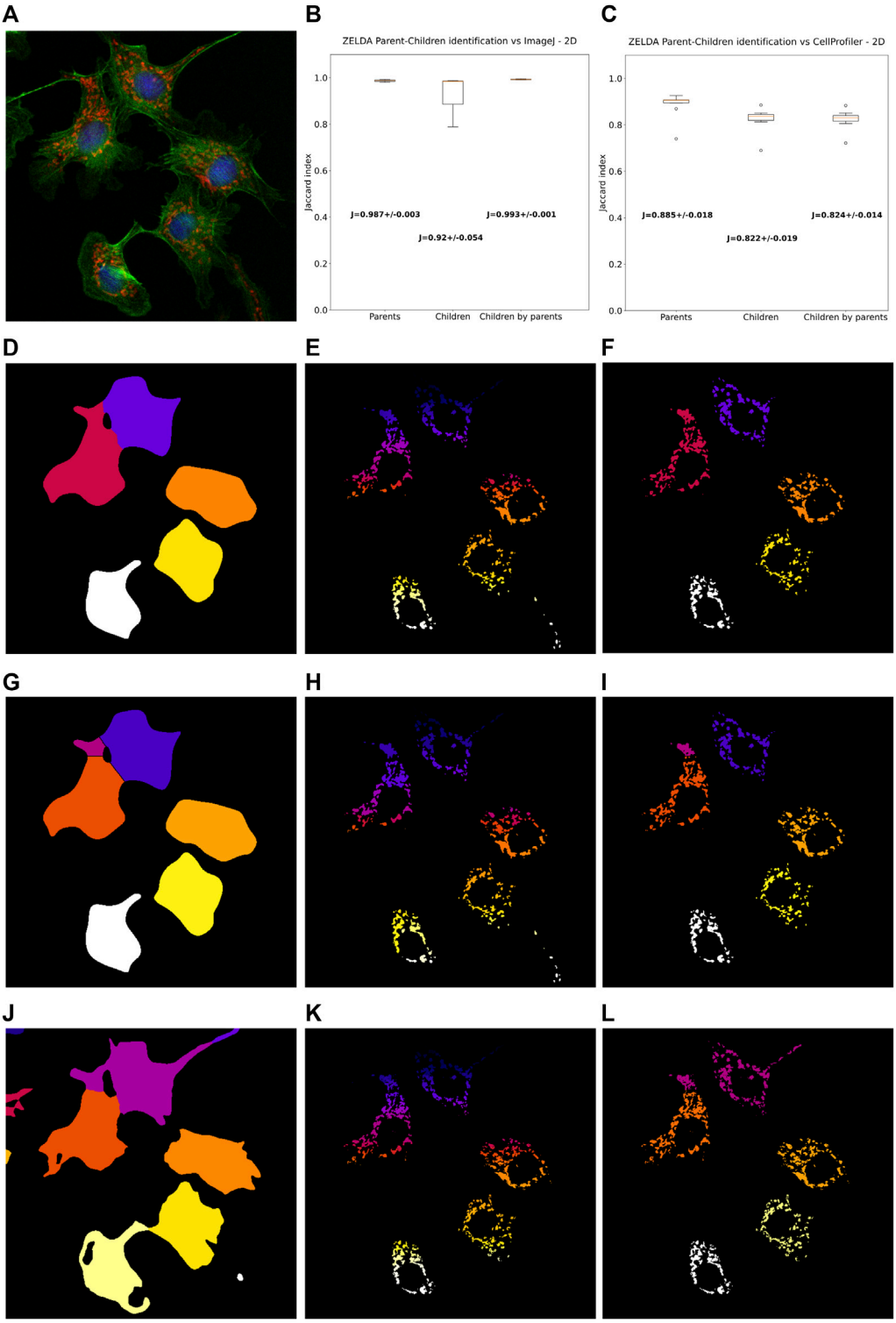

FIGURE 4 | ZELDA 2D segmentation and "parent-child" relation benchmarked with *ImageJ* and *CellProfiler*. **(A)** 2D images of BPAE cells stained with DAPI (blue, cell nuclei), AlexaFluor 488 (green, cytoplasms), and MitoTracker Red (red, mitochondria). **(B)** Labelling comparison between ZELDA and *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

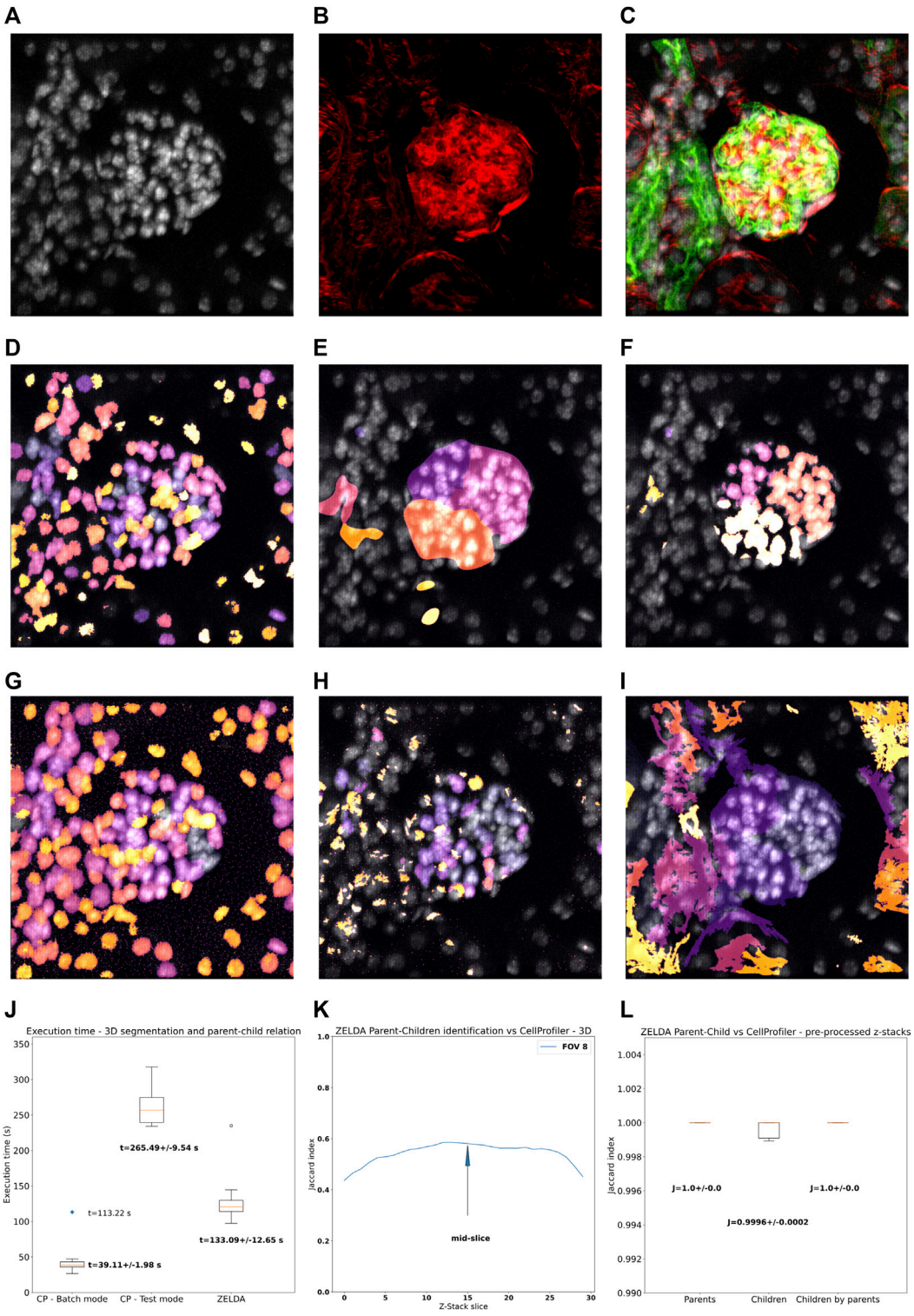

FIGURE 5 | 3D segmentation, "parent-child" relation, and execution time of ZELDA compared to *CellProfiler*. Z-stacks of mouse kidney tissue showing glomeruli were used for the benchmark in 3D. **(A)** DAPI staining used to segment the cell nuclei. **(B)** Phalloidin used to identify the glomerular structures **(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.