MorphData: Automating the data extraction process of morphological features of microglial cells in ImageJ

Ana Bela Campos‡ *[a,b], Sara Duarte-Silva [a,b], António Francisco Ambrósio [c,d,e], Patrícia Maciel [a,b], Bruno Fernandes‡ [f]

[a] Life and Health Sciences Research Institute (ICVS), School of Medicine, University of Minho, Braga, Portugal

[b] ICVS/3B's, PT Government Associate Laboratory, Braga/Guimarães, Portugal

[c] University of Coimbra, Coimbra Institute for Clinical and Biomedical Research (iCBR), Faculty of Medicine, Coimbra, Portugal

[d] University of Coimbra, Center for Innovative Biomedicine and Biotechnology (CIBB), Coimbra, Portugal

[e] Clinical Academic Center of Coimbra (CACC), Coimbra, Portugal

[f] Department of Informatics, ALGORITMI Center, University of Minho, Braga, Portugal

‡ Contributed equally to the article.

* To whom correspondence should be addressed.

Ana Bela Campos. Life and Health Sciences Research Institute (ICVS), School of Medicine, University of Minho, Braga, Portugal. ana.bela.campos.88@gmail.com

Keywords: Bioinformatics, Cell Biology, Data Extraction, Imaging, Microglia, Morphology.

Abbreviations: CTR: control; EX: experimental group; GUI: Graphical User Interface; IJM: ImageJ Macro language; PBS: Phosphate-Buffered Saline; RT: room temperature.

Abstract

Microglial cells are the first line of defense within the central nervous system, with morphological characterization being widely used to define their activation status. Most methods to evaluate microglia status are manual, and, therefore, often biased, inaccurate, and time consuming. In fact, the process to collect morphological data starts with the acquisition of photomicrographs from where images of single cells are extracted. Then, the researcher collects the morphological features that characterize each cell. However, a manual data collection process from single cells can take weeks to complete. This work describes an open-source ImageJ plugin, *MorphData*, which automatizes the data extraction process of morphological features of single microglial cells. The plugin collects, processes, and organizes features associated with cell complexity and ramification. In a computer with limited computing power, it took less than 14 minutes to handle the morphological features of 699 single cells of two experimental groups. The same process, if performed manually, would take around 19 working days. Overall, *MorphData* significantly reduces the time taken to collect morphological data from microglial cells, which can then be used to study, understand, and characterize microglia behavior in the brain of human patients or of animal models of neurological and psychiatric diseases.

1. Introduction

Microglial cells represent a population of macrophages-like cells in the central nervous system, with a broad range of roles in neurodevelopment, synaptic plasticity, and brain protection and repair [1]. Hence, the morphological characterization of these cells is of the utmost importance to ascertain and establish their state in particular conditions. It is known that microglia morphology and function are closely related [2]. In fact, in response to injury, microglial cells undergo morphological and functional changes, changing from a highly ramified into an amoeboid-like shape [3]. This implies that a rigorous analysis of microglia morphology data is of essence for the understanding of cellular behavior [3, 4, 5].

The collection of morphological data goes through several steps. First, one is required to obtain photomicrographs from where images of cells can be extracted. Then, ImageJ is required for image processing [6]. Being an open-source software, it is frequent to find macros and plugins, conceived by the community, that provide ImageJ with extra features. Examples include *SlideJ* [7], *aNMJ-morph* [8], and *AutoNeuriteJ* [9], among others. A different example is provided by Heindl and colleagues [10], where the authors developed a method outside ImageJ, using a proprietary programming language and numeric computing environment.

For the morphological analysis of microglial cells, ImageJ provides two key plugins: (i) *AnalyzeSkeleton (2D/3D)* [11], which tags skeletal features relevant to cell ramification, and (ii) *FracLac* [12], which quantifies cell surface and size, soma thickness, and the cylindrical shape of cells. The use of both plugins is recommended, as cell ramification data are complementary to cell complexity [13]. While the former may be applied to entire photomicrographs, the latter is applied over single cells, thus requiring a significant investment of time to collect all data. However, since the former is typically applied over entire photomicrographs, results come with significant noise. Hence, we aimed to develop a protocol that allowed the application of both plugins over single cells, not only reducing the amount of noise that comes from analyzing larger and noisy photomicrographs, but also solving the problem of stacked cells. This comes, however, with a significant time cost when collecting the morphological features that characterize each single cell. In fact, the process to obtain morphological features, when performed manually over each cell, is a demanding, repetitive, and laborious task, that can take several weeks to complete. Another potential issue is the human error associated with the data collection process. This sets the need for the *MorphData* plugin.

This manuscript describes the design, implementation, and use of a new plugin that automatically runs and collects morphological features for single cells in a matter of minutes, significantly reducing the time spent on the data collection process. The goal of *MorphData* is set on optimizing the data collection process of morphological features.

2. Materials and methods

2.1 Ethics statement

All procedures with mice were conducted in accordance with the ARRIVE 2.0 guidelines (Animal Research: Reporting *In Vivo* Experiments). Animal facilities and the people that worked directly in animal procedures were certified by the Portuguese regulatory – *Direção Geral de Alimentação e Veterinária*, license number 020317. All animal procedures were approved by the Animal Ethics Committee of the Life and Health Sciences Research Institute, University of Minho (SECVS 120/2014), and conducted in consonance with the European Union Directive 2010/63/EU. Health monitoring was performed according to the Federation of European Laboratory Animal Science Associations guidelines, where the Specified Pathogen Free health status was confirmed by sentinel mice maintained in the same animal housing room.

2.2 Animal maintenance

Two groups, control (CTR) and experimental group (EX) mice on a C57BL/6J background, were considered. Animals were maintained in a conventional animal facility and under standard laboratory conditions, which includes an artificial 12h light/dark cycle, lights on from 8:00 am to 8:00 pm, an ambient temperature of 21 ± 1ºC and relative humidity of 50–60%.

2.3 Immunofluorescence staining

CTR (n=4) and EX (n=4) mice were deeply anesthetized with a mixture of ketamine hydrochloride (150mg/kg) and medetomidine (0.3mg/kg), and transcardially perfused with phosphate-buffered saline (PBS) followed by 4% paraformaldehyde (PFA) solution (PFA, 0.1 M, pH 7.4, in PBS). Brains were removed and immersed in 4% PFA (48h), followed by 1 week in a 30% sucrose PBS buffer (at 4ºC). Coronal sections were obtained using a vibratome (VT1000S, Leica, Germany) with 40µm of thickness. For staining, the permeabilization in the free-floating sections was performed with PBS-T 0.3% (0.3% triton X-100, Sigma Aldrich, in PBS) for 10 min, followed by immersing the slices in pre-heated citrate buffer (10 mM, pH 6.0; Sigma Aldrich) during 20 min using a thermoblock (D1200, LabNet) set at 80ºC. Once cooled, slices were blocked with goat serum blocking buffer (10% normal goat serum, 0.3% triton X-100, in PBS) at room temperature (RT) for 90 min. After this, the sections were incubated with the primary antibody anti-ionized calcium binding adaptor molecule 1 (rabbit polyclonal IgG anti-Iba-1, 1:600; Wako) overnight at 4ºC. In the next day, sections were incubated with a secondary antibody (Alexa Fluor 594 goat anti-rabbit, 1:1000; ThermoFisher Scientific) during 90 min at RT, protected from light, followed with 4',6-Diamidin-2-phenylindol (DAPI, 1:1000; Invitrogen) for nuclei staining. Sections were mounted on microscope slides (Menzel-Glaser Superfrost©Plus, Thermo Fisher Scientific) and covered with a coverslip (Menzel-Glaser 24–60mm, Wagner und Munz) using aqueous mounting medium (Fluoromount TM, Sigma-Aldrich).

2.4 Image acquisition

Images were acquired using the Olympus Confocal FV1000 laser scanning microscope, with a resolution of 1024×1024px, using a 40× objective (UPlanSApo, N.A. 0.90; dry; field size 624.39×624.39μm; 0.31μm/px), being used to obtain Z-stacked images, which include two distinct channels (red, Iba-1; blue, DAPI). The acquisition settings were the following: scanning speed = 4μm/px; pinhole aperture = 110μm; Iba-1, excitation = 559nm, emission = 618nm; DAPI, excitation = 405nm, emission = 461nm; in a 3-dimensional scenario (X, Y, and Z axis). Four coronal brain sections per animal were imaged in both hemispheres, for a particular region of interest, to yield 4-6 digital photomicrographs per section containing the region of analysis.

2.5 Image preparation

The Z-stacked 3D volume images from sections of the brain region of interest were prepared for microglial morphology analysis using a semi-automatic method, adapted from [13], to obtain both skeleton and fractal data. However, contrary to the cited method, we went further and obtained binary (white cells on black background) single cells (one cell per file) to feed both the *AnalyzeSkeleton (2D/3D)* and the *FracLac* plugins.

Briefly, after stacking the 3D volume images, the double-color image was split to obtain the Iba-1 label in the red channel, which accurately mirrors the cell profile. Brightness and contrast of the red-channel were adjusted as needed and an unsharp mask was applied. Then, a despeckle filter was used to remove salt and pepper noise, with the threshold option being used and adjusted, as needed, to convert the image into a binary one. Noise was subsequently reduced using despeckle and by removing outliers. After that, random cells from both the original and the binary images were selected with the rectangle tool, using the region of interest to set the same rectangle dimensions for all the selected cells (field size 296×264). Then, after selecting the cells, the paintbrush tool was used to complete and draw the morphology of the cells (always comparing with the original image) and to clean extra signal that is not related to these cells, thus producing a single-cell image without any noise. This process is summarized in Figure 1A and 1B.

699 single-cell images, for both CTR (310 single cells) and EX (389 single cells) groups, were obtained and stored in the file system, in the TIFF format. At this point, the researcher is ready to start collecting the morphological features that characterize each microglial cell using the *MorphData* plugin.

2.6 Experimental setup

The *MorphData* plugin requires basic computational resources. The experiments here described were carried out on a personal computer with an 8th generation i7 CPU with 4 cores at 1.80GHz, 8GB of RAM, a SSD disk, and the Windows 10 operating system. ImageJ 1.53c, embedded in

Fiji, has been set with 6989MB of maximum heap size. For scripting, Python, version 3.7.10, was used, together with a set of libraries, including pandas, math, and tkinter.

2.7 Requirements

To successfully execute the *MorphData* plugin, the user must comply with a set of requirements:

i. ImageJ version must be, at least, 1.52t;
ii. *AnalyzeSkeleton (2D/3D)* plugin installed in ImageJ;
iii. *FracLac* plugin installed in ImageJ;
iv. Collect a set of binary single cells from where morphological data will be obtained.

It is important to clearly structure the obtained images in the file system. Ideally, the user should create a structure such as the one depicted in Figure 2A. To comply with the *MorphData* plugin, while the name of the folders at the two first levels is irrelevant, it is important to guarantee that the last two levels are entitled as "*Slice i*", where *i* identifies different slices, and "*Image j*", where *j* identifies different photomicrographs. Single cells should be placed inside the corresponding image folder, being entitled as "*Microgliak*.tif", where *k* identifies each cell within the image folder.

3. Results and discussion

3.1 Implementation

The *MorphData* plugin was developed using ImageJ Macro language (IJM), a scripting language that allows a developer to control many features of ImageJ. Plugins written in IJM can be programmed to perform sequences of actions, thus automating repetitive processes. It has a set of basic structures, including variables, conditional statements, loops, and user-defined functions. Importantly, IJM allows the developer to access ImageJ functions that are available from its Graphical User Interface (GUI). *MorphData* takes advantage of IJM to automatically collect morphological features, working on any operating system in which ImageJ can work. The plugin is open-source and available online (*github.com/anabelacampos/MorphData*). A straightforward architectural diagram is depicted in Figure 2B.

3.2 MorphData algorithm

The *MorphData* plugin, when executed, starts by asking the user to indicate the folder containing the single cell images (Figure 3A). Following the file structure defined in Figure 2A, the user should indicate the *EX* folder (the root folder). The plugin then creates auxiliary folders to store the collected data and starts navigating the indicated folder looking for single cell images. Then, for each image, the algorithm is summarized as follows:

i. To obtain skeletal features relevant to cell ramification:
   a) Open a single cell;
   b) Run the command “Process > Binary > Skeletonize” to create a skeletonized image;
   c) Run the *Analyze Skeleton (2D/3D)* plugin;
   d) Run the “saveAs” command to collect and store, in a csv file, skeletal data;
   e) Run the “saveAs” command to collect and store, in a csv file, branch information data.
ii. To obtain features relevant to cell complexity:
   a) Open a single cell;
   b) Run the “saveAs” command to store a shaped single cell, in TIFF format, in a folder entitled as “Area”;
   c) Run the “Process > Binary > Outline" and “saveAs” commands to store an outlined single cell, in TIFF format, in a folder entitled as “Perimeter”.
iii. Repeat steps 1. and 2. for each single cell;
iv. At the end, the algorithm indicates the number of analyzed cells.

Finally, contrary to the *Analyze Skeleton (2D/3D)* plugin, which is automatically executed by *MorphData*, the *FracLac* plugin cannot be directly executed from within another plugin. This limitation requires the user to manually execute the *FracLac* plugin itself after the *MorphData* plugin has finished. Fortunately, since the "Area" and the "Perimeter" folders, which were automatically created by *MorphData*, already contain all shaped and outlined cells (Figure 3B and 3C), the user can execute the *FracLac* plugin in batch mode. Hence, with a single execution of this plugin, the user obtains fractal data for all cells almost immediately (avoiding the need to execute *FracLac* for each cell individually).

3.3 Post-processing

Up to this point, all morphological data are now available, for all single cells, in multiple csv files in auxiliary "results" folders. In total, the *MorphData* plugin gathers 221 features (from skeleton to fractal ones), and some of them may be irrelevant to the characterization of microglial cells. Hence, the post-processing step consists in joining all data, cleaning irrelevant features, and performing a feature engineering process to create new features, including the *cell_area*, *cell_perimeter*, *roughness*, and *cell_circularity*, among others.

Due to the potential high number of rows (cells) and columns (morphological features), an ImageJ plugin is unsuitable for the task, as it would eventually run out of memory. Hence, a Python script, entitled as *MorphData_PostProcessing.py*, was conceived and released as part of the *MorphData* plugin. This script requires a simple python environment to execute, again asking the user to indicate the location of the root folder. It will then automatically apply the post-processing procedures, creating three final files, containing the following 46 features:

i. **skeleton_final_results.csv:**
   - *# Branches, # Junctions, # End-point voxels, # Junction voxels, # Slab voxels, Average Branch Length, # Triple points, # Quadruple points, Maximum Branch Length, animal, microglia_id*.

ii. **branch_info_final_results.csv:**
   - *Skeleton ID, Branch length, V1 x, V1 y, V1 z, V2 x, V2 y, V2 z, Euclidean distance, running average length, average intensity (inner 3rd), average intensity, animal, microglia_id*.

iii. **fraclac_final_results.csv:**
   - *fractal_dimension, lacunarity, outline_mean_fg, density, span_ratio_major_minor, convex_hull_area, convex_hull_perimeter, convex_hull_circularity, diameter_bounding_circle, mean_radius, max_span_across_convex_hull, max_min_radii, shape_mean_fg, 1_pixel_side_micron, 1_pixel_area_micron_sq, cell_area, cell_perimeter, roughness, cell_circularity, animal, microglia_id*.

Figure 3D contains a graphical perspective of part of the content of the *skeleton_final_results.csv* file, which contains 11 features relevant for cell ramification. The remaining two files are similar, varying only on the quantified features.

3.4 Performance evaluation

The performance of the *MorphData* plugin and its post-processing script was evaluated based on the validity of the collected values and on the time it took to obtain the morphological data of all single cells of both experimental groups, when compared to a manual collection of such data.

Totally, in a computer with limited computing power, it took less than 14 minutes to collect 46 morphological features associated with 699 single cells of two experimental groups. In particular, 6.5 minutes were spent by the *MorphData* plugin, and its post-processing script, handling the CTR group. Of those, nearly 3 minutes were spent collecting skeleton data, 3.25 minutes by the *FracLac* plugin on batch mode, and 10 seconds by the post-processing script. On the other hand, 7.5 minutes were spent handling the EX group. Of those, 3.5 minutes were spent collecting skeleton data, 3.8 minutes by the *FracLac* plugin on batch mode, and 11 seconds by the post-processing script.

The same process was performed manually, by a skilled user of ImageJ, for a set of ten single cells of the CTR group. To ease the process, the same file system structure (as required by the *MorphData* plugin) was used. The goal was to mimic the processes that are automatically performed by *MorphData*, and manually collect 46 morphological features for the ten cells. The mean time to collect such morphological features was of 13 minutes per cell. Skeleton data were faster to collect (around 1.5 minutes), since the *AnalyzeSkeleton (2D/3D)* plugin only opens two results' windows that the user can immediately save in two distinct files in the file system, in csv format, and then close the opened windows. However, fractal data were considerably harder to collect (around 11.5 minutes). On the one hand, for each cell, the *FracLac* plugin must be executed twice - one for a shaped cell and one for an outlined cell, which the user must prepare. On the other hand, for each execution, this plugin opens multiple results' windows. The ones to keep opened are the “Hull and Circle Results” and the “Box Count Summary” windows. However, these results' windows are not user-friendly and, besides providing the user with an overwhelming amount of 173 features (most of them formulas and unwanted columns), it does not allow the user to copy only the desired features - the user must manually write each value of each desired feature to a csv, or excel, file. In fact, the process of selecting features from the *FracLac* plugin is extremely exhausting and error-prone. Finally, it is up to the user to calculate the value of non-existing features such as *cell_area*, *cell_perimeter*, *roughness*, and *cell_circularity*.

A couple more obstacles emerged with the manual process. First, the user was required to edit each stored file, for each cell, to identify the animal and the microglia of each row of data. Secondly, the user was required to copy the contents of each file to an overall file, aggregating the data for the experimental group - since each cell is made of three files (two skeleton files and one fractal file) this would require the user to open and copy 699×3 files, which would, again, be a time-consuming task that would have to be performed after collecting all data.

Overall, the manual process for ten single cells of the CTR group took more than 2 hours to complete. On the other hand, the *MorphData* plugin and its post-processing script took less than 14 minutes to collect, process, and organize the morphological features of 699 cells. Assuming a mean value of 13 minutes per cell, the manual process to collect the morphological features of all cells would take 151 hours, which corresponds to almost 19 working days (8 hours/day) collecting data without stopping.

MorphData brings obvious advantages, mainly by significantly reducing the time it takes to collect morphological data. These values could be further reduced by a computer with higher computation power. In addition, the automation of the data collection process completely removes the risk of human error. It is worth mentioning that since *MorphData* is using well established plugins to collect morphological features, it produces the same exact results as when performing the data collection process manually. In fact, *MorphData*'s collected values were further compared and validated with multiple cells data that were manually collected, without a single collection error.

Finally, some limitations must be acknowledged. *MorphData*'s focuses on microglial cells. Hence, the plugin may require adjustments for different cell types. The plugin was also conceived to work with binary white cells on a black background. The plugin can, however, be modified to work with the reverse.

4. Concluding Remarks

Morphological characterization of cells is highly relevant in the sciences field, and particularly in neurosciences. However, when performed manually, the process for obtaining morphological features from single cells is a demanding, repetitive, and laborious task; it is error-prone and can take several weeks to complete.

MorphData automatizes the data extraction process of morphological features of microglial cells. This plugin has already been used successfully in a morphological characterization study, where several thousands of single cells, from multiple experimental groups, were used. The benefits were considerable - several weeks of work were spared. Here, we demonstrate the ability of the *MorphData* plugin and its post-processing script to handle 699 microglial cells from two experimental groups. It took the plugin 14 minutes to collect, process, and organize 46 morphological features. If performed manually, this would take 19 working days just to collect and process the data.

The conceived macros, python scripts, used data, and other examples can be found at *MorphData*'s online repository, under an open-source license.

Acknowledgements

This work is supported by *Fundação para a Ciência e a Tecnologia* (FCT) (PTDC/NEU-NMC/3648/2014) and COMPETE-FEDER (POCI-01-0145-FEDER-016818). ABC is supported by a doctoral fellowship from FCT (PD/BD/127828/2016).

Authors Contributions

ABC: Conceptualization; Methodology; Validation; Formal analysis; Investigation; Data Curation; Writing - Original Draft; Writing - Review & Editing; Visualization. SDS: Conceptualization; Resources; Writing - Review & Editing; Supervision. AFA: Conceptualization; Resources; Writing - Review & Editing; Supervision; Funding acquisition. PM: Conceptualization; Resources; Writing - Review & Editing; Supervision; Funding acquisition. BF: Methodology; Software; Validation; Formal analysis; Data Curation; Writing - Review & Editing; Visualization

Conflicts of interest

The authors declare no commercial or financial conflict of interest.

References

[1] S. Bachiller, I. Jimenez-Ferrer, A. Paulus, Y. Yang, M. Swanberg, T. Deierborg, A. Boza-Serrano, *Front. Cell. Neurosci.* **2018**, *12*, 488. DOI: 10.3389/fncel.2018.00488.

[2] M. Fernandez-Arjona, J. Grondona, P. Granados-Duran, P. Fernandez-Llebrez, M. Lopez-Avalos, *Front. Cell. Neurosci.* **2017**, 11, 235. DOI: 10.3389/fncel.2017.00235.

[3] H. Morrison, J. Filosa, *J. Neuroinflammation* **2013**, 10, 782. DOI: 10.1186/1742-2094-10-4.

[4] R. Kongsui, S. Beynon, S. Johnson, F. Walker, *J. Neuroinflammation* **2014**, 11, 182. DOI: 10.1186/s12974-014-0182-7.

[5] B. Davis, M. Salinas-Navarro, M. Cordeiro, L. Moons, L. Groef, *Sci. Rep.* **2017**, 7, 1576. DOI: 10.1038/s41598-017-01747-8.

[6] C. Schneider, W. Rasband, K. Eliceiri, *Nat. Methods* **2012**, 9, 671. DOI: 10.1038/nmeth.2089.

[7] V. Mea, G. Baroni, D. Pilutti, C. Loreto. *PLoS One* **2017**, 12, e0180540. DOI: 10.1371/journal.pone.0180540.

[8] G. Minty, A. Hoppen, I. Boehm, A. Alhindi, L. Gibb, E. Potter, B. Wagner, J. Miller, R. Skipworth, T. Gillingwater, R. Jones, *R. Soc. open sci.* **2020**, 7, 200128. DOI: 10.1098/rsos.200128.

[9] B. Boulan, A. Beghin, C. Ravanello, J. Deloulme, S. Gory-Faure, A. Andrieux, J. Brocard, E. Denarier, *PLoS One* **2020**, 15, e0234529. DOI: 10.1371/journal.pone.0234529.

[10] S. Heindl, B. Gesierich, C. Benakis, G. Llovera, M. Duering, A. Liesz, *Front. Cell. Neurosci*. **2018**, 12, 106. DOI: 10.3389/fncel.2018.00106.

[11] I. Arganda-Carreras, R. Fernandez-Gonzalez, A. Munoz-Barrutia, C. Ortiz-De-Solorzano, *Microsc. Res. Tech.* **2010**, 73, 1019. DOI: 10.1002/jemt.20829.

[12] A. Karperien, *FracLac for ImageJ*, **1999-2013**. Available online at: http://rsb.info.nih.gov/ij/plugins/fraclac/FLHelp/Introduction.htm.

[13] K. Young, H. Morrison, *J. Vis. Exp.* **2018**, 136, e57648. DOI: 10.3791/57648.

Figures

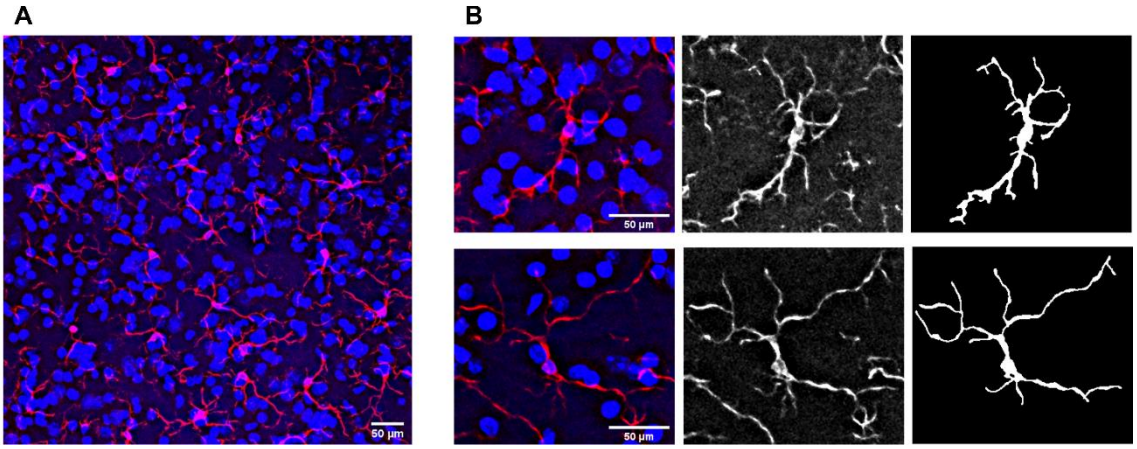


Figure 1. Representative photomicrograph from the region of interest, showing microglial cells (red) and the representation of two binary single microglial cells (gray) from a control mouse. (A) Original Z-stacked 3D volume photomicrograph. (B) Going from a noisy cell to a binary single cell (from the left to the right).

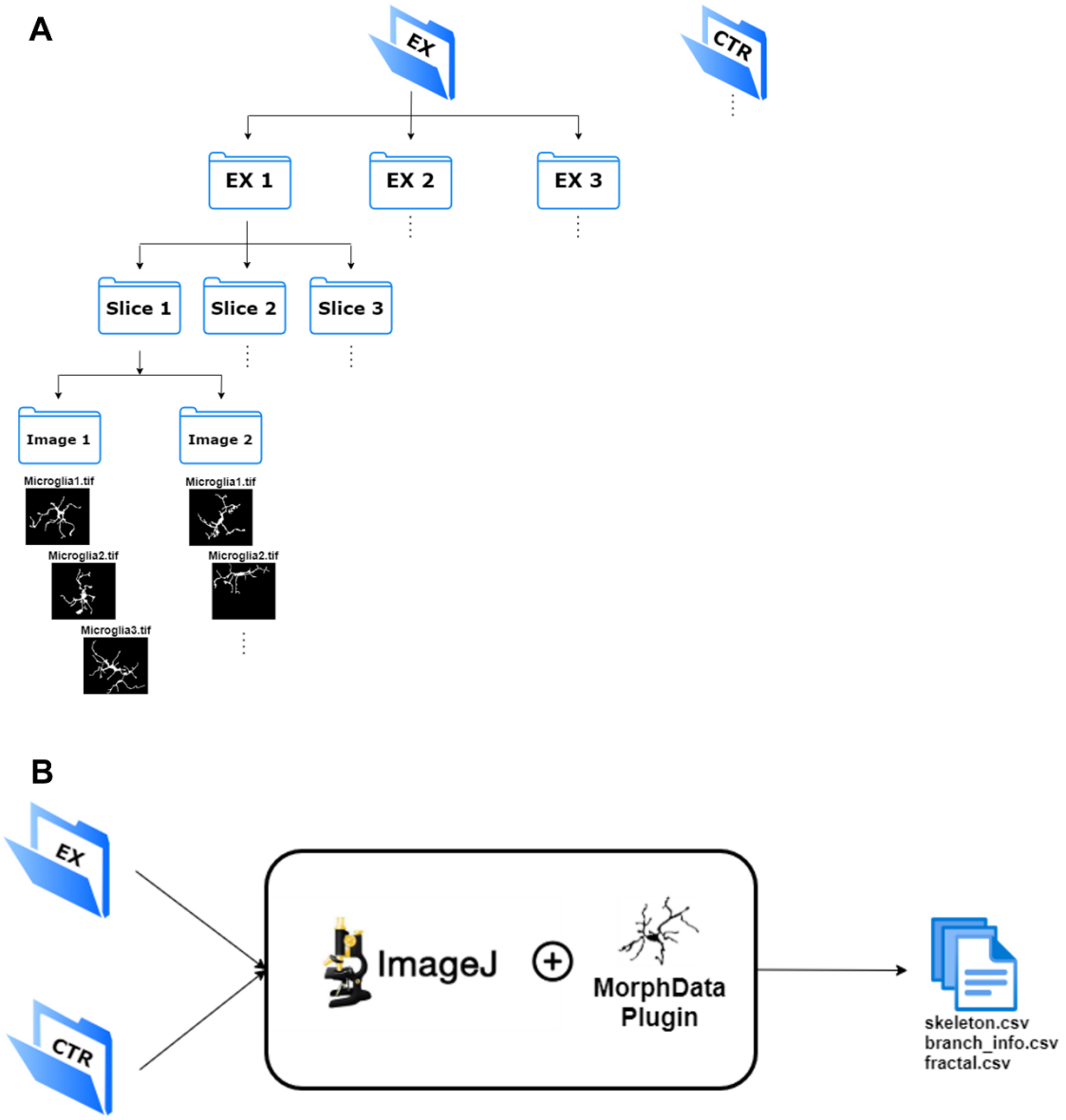
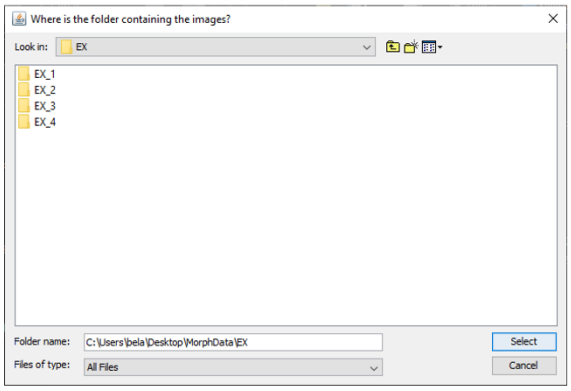
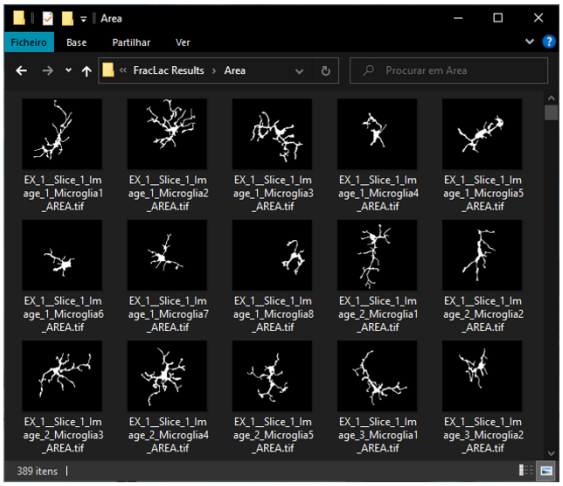


Figure 2. *MorphData*'s architectural structure. (A) Recommended file system structure to store single cells images. The root folders, *EX* and *CTR*, hold the images of the corresponding experimental group. (B) *MorphData*'s architectural diagram, receiving, as input, the root folders, and producing, as output, three csv files with the morphological features that characterize each single cell.

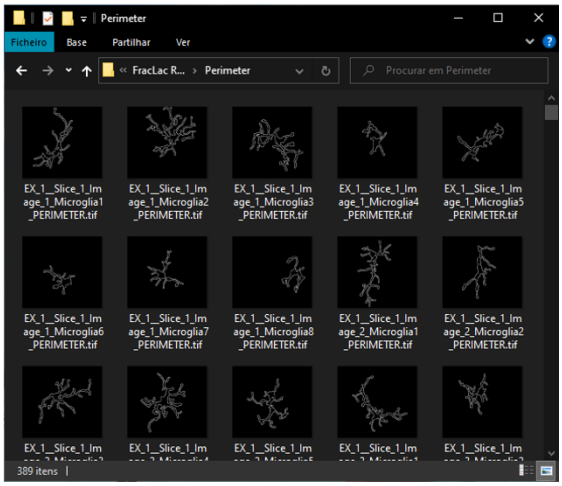
A



B



C



D

# Branches	# Junctions	# End-point voxels	# Junction voxels	# Slab voxels	Average Branch	# Triple points	# Quadruple points	Maximum Branch	animal	microglia
37	18	17	60	632	13.826	15	3	57.22	EX_1	EX_1_Slice_1_Image_1_Microglia1
62	30	31	93	875	11.599	27	3	52.533	EX_1	EX_1_Slice_1_Image_1_Microglia2
58	27	31	80	783	10.837	23	4	49.859	EX_1	EX_1_Slice_1_Image_1_Microglia3
20	9	12	33	279	11.787	8	1	39.284	EX_1	EX_1_Slice_1_Image_1_Microglia4
25	12	13	41	454	14.73	11	1	44.206	EX_1	EX_1_Slice_1_Image_1_Microglia5
14	6	9	23	268	15.275	5	1	27.674	EX_1	EX_1_Slice_1_Image_1_Microglia6
18	8	11	27	359	15.946	7	1	30.975	EX_1	EX_1_Slice_1_Image_1_Microglia7
24	12	12	41	288	10.021	10	1	56.567	EX_1	EX_1_Slice_1_Image_1_Microglia8
41	19	22	66	667	13.024	17	1	37.073	EX_1	EX_1_Slice_1_Image_2_Microglia1
15	7	9	24	435	21.661	7	0	46.428	EX_1	EX_1_Slice_1_Image_2_Microglia2

Figure 3. Execution and results of the *MorphData* plugin. (A) *MorphData* dialog GUI asking the user where the single cell images are located. (B) Shaped images, produced by *MorphData*, stored in the "Area" folder. (C) Outlined images, produced by *MorphData*, stored in the "Perimeter" folder. Both shaped and outlined images are ready to be passed to the *FracLac* plugin for batch mode execution. (D) A sample of the *skeleton_final_results.csv* file, produced by *MorphData*. This file contains 11 features relevant to cell ramification and cell identification.