A Fiji pipeline to segment 3D objects and retrieve shape parameters in biomedical images: "Amyloid XPCT Workflow"

Modus operandi, version 1, March 2021

Matthieu Chourrout, Margaux Roux, Coralie Gislard, Ignacio Arganda-Carreras, David Legland, Hugo Rositi, Marlène Wiart, Fabien Chauveau

Corresponding author:
Matthieu CHOURROUT
Lyon Neuroscience Research Center
Groupement Hospitalier Est, Bât. CERMEP-Imagerie du Vivant
59 BD PINEL, 69677 BRON cedex
FRANCE
Mail. matthieu.chourrout@univ-lyon1.fr

Introduction

Biologists often lack image processing tools that are easy to understand and to adjust to their specific problem. With that in mind, we established a detailed modus operandi, based on free open-source programs. It involves a trainable algorithm: this kind of algorithms are best suited to extract elements that are not easily distinguishable (including non-specific signal intensity, varying textures, etc.).

In our case, it was designed to assess the morphology of amyloid-β plaques in transgenic mice, which are rodent models for Alzheimer's disease. The images were acquired as X-Ray Phase Contrast Tomography (XPCT) on synchrotron beamlines. Our Fiji pipeline was built with the following plugins: segmentation editor (to isolate hippocampus), trainable WEKA segmentation 3D (to identify plaques), MorpholibJ and 3D ImageJ suite (to label objects and extract relevant shape parameters). Since these plugins were not developed for this specific application, the present pipeline is likely to be well-suited for any morphometric analysis of small 3D objects.

Prerequisite: installation of the tools

ImageJ is used to perform simple pre-processing and as the core element for the pipeline; we chose to use the most furnished distribution named Fiji: https://fiji.sc/

The trainable algorithm is WEKA (Waikato Environment for Knowledge Analysis), a project of a research team from the University of Waikato. It is a standalone software. You have to install it on the computer from its official site: https://waikato.github.io/weka-wiki/downloading_weka/

Then WEKA will be called from ImageJ thanks to the plugin "Trainable WEKA Segmentation" available at https://imagej.net/Trainable_Weka_Segmentation. Besides, you can improve feature extraction (structures, derivatives, etc.) by installing the ImageScience plugin from https://imagej.net/ImageScience.

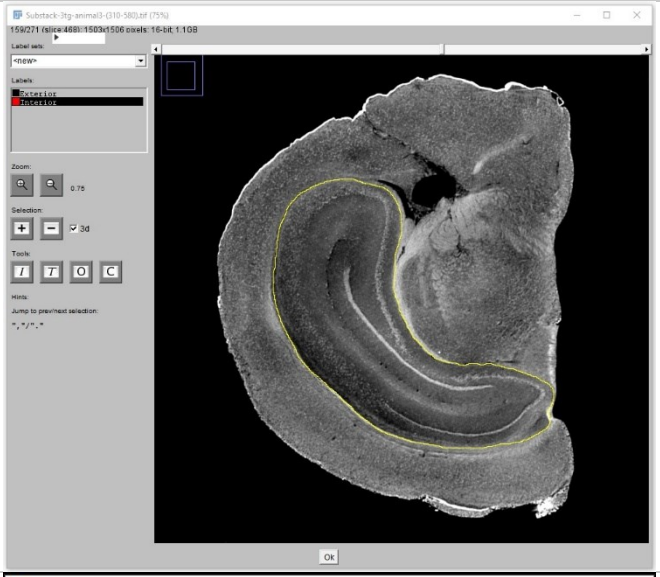
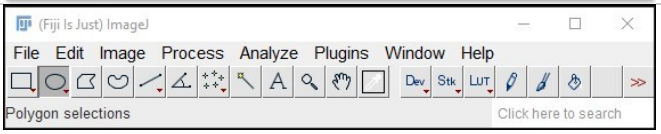
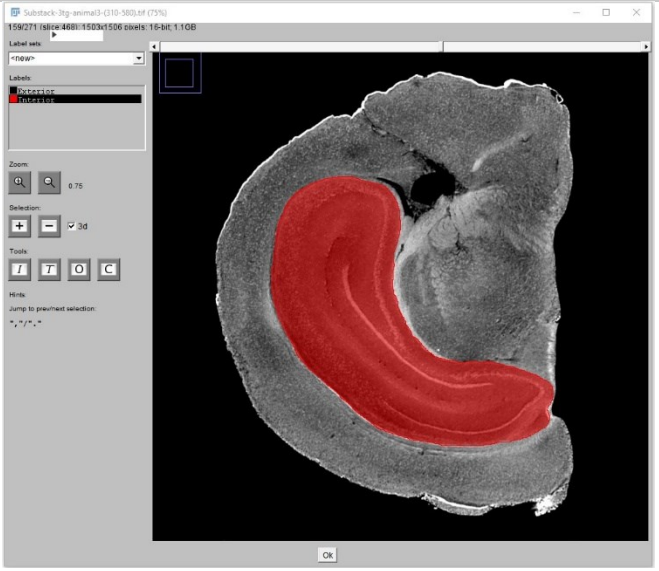
The post-processing is done with the MorphoLibJ plugin: https://imagej.net/MorphoLibJ; it labels the images and provides tools to filter the objects. More metrics are available through the 3D ImageJ Suite plugin (including a more accurate sphericity for small objects): https://imagej.net/3D_ImageJ_Suite.

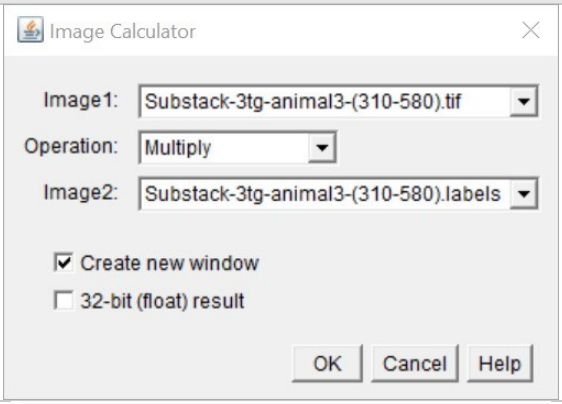
You are ready to apply the pipeline once you have the following tools:

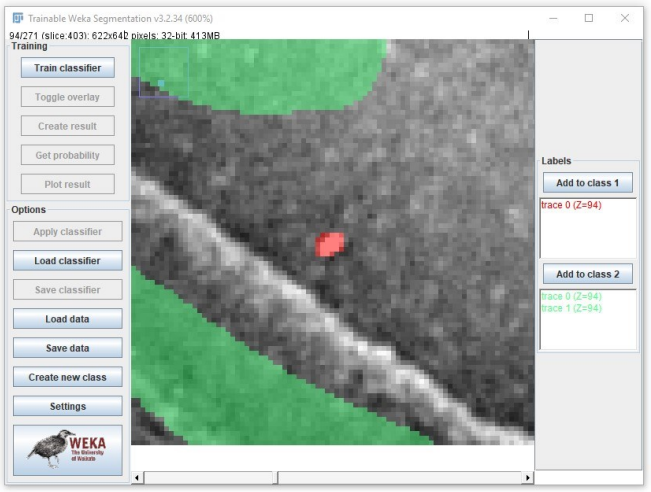
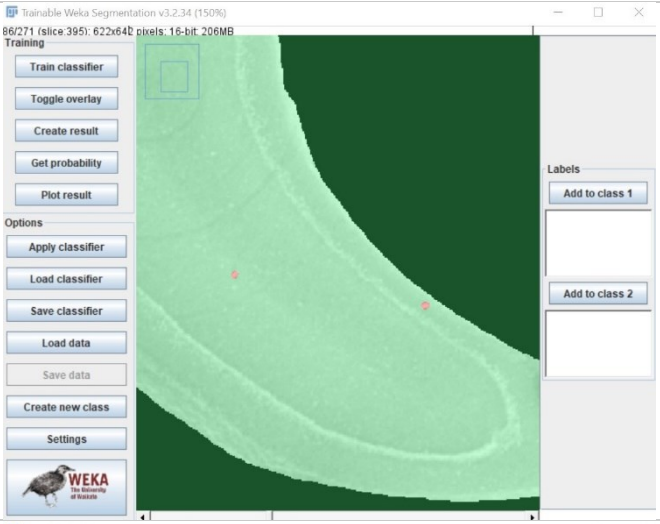
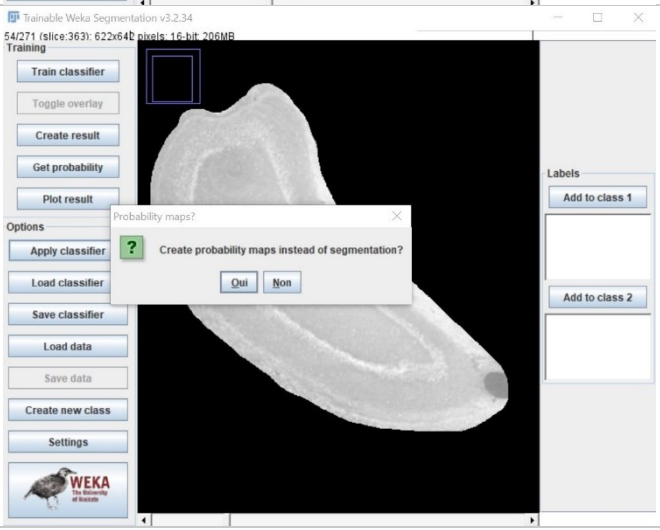
- The Fiji distribution of ImageJ, along with the following plugins:
    - Trainable WEKA Segmentation,
    - ImageScience (optional),
    - MorphoLibJ,
    - 3D ImageJ Suite (optional);
- The WEKA software.

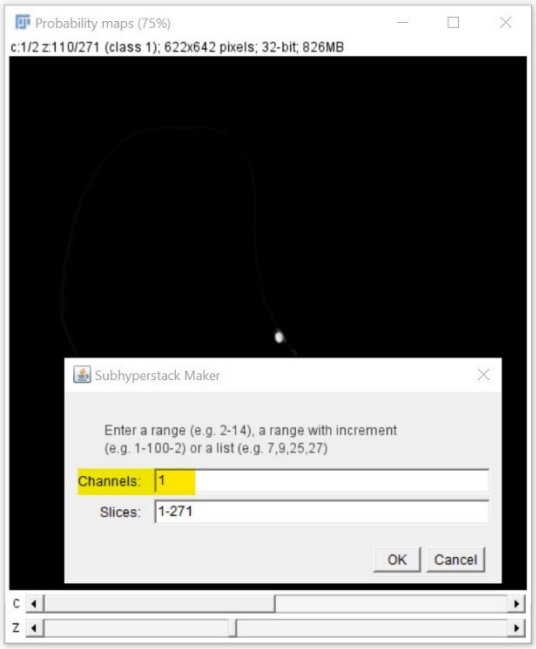
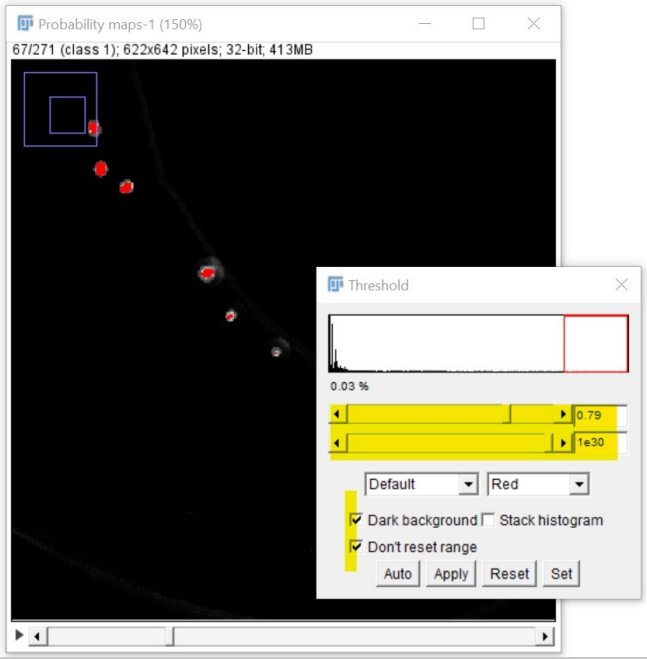
We performed processing on a computation station with the following specifications: twin-processor (Intel® Xeon® Gold 6140M @ 2.30 GHz) CPU (36 cores), 512 Go RAM @ 2.7 GHz. For reference, the training described at step 4.1 (using 3 feature metrics) was completed within 10 min on a 200 Mo segmented mouse brain hippocampus. An example of a resulting 3D rendering is provided as the video file "AmyloidXPCT-3Dviewer", and shows the whole mouse brain, with hippocampus pre-segmented, and amyloid plaques segmented inside the hippocampus).

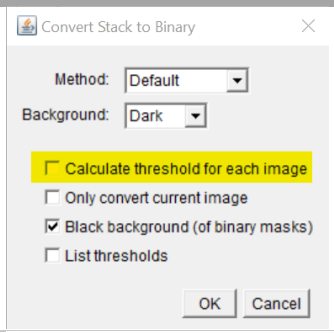
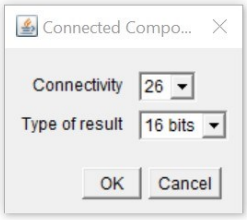
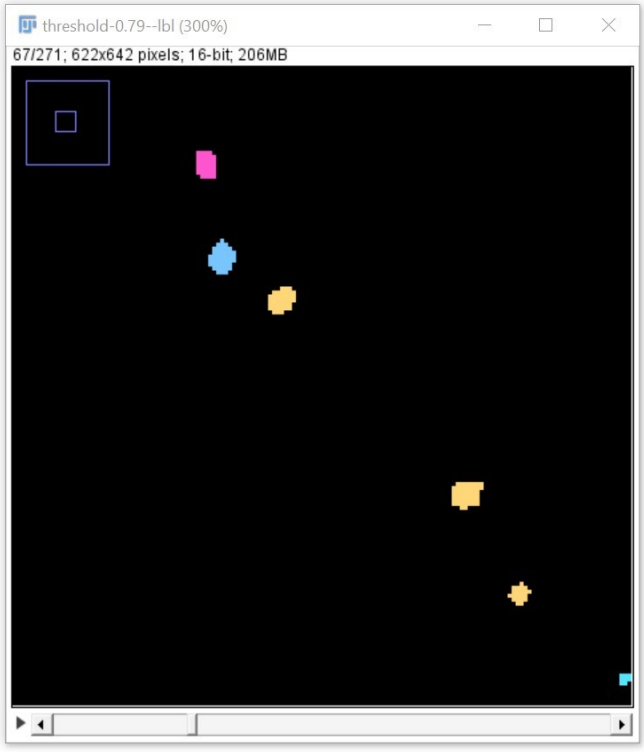
Detailed workflow

#	Step / Actions	Illustrations
1	**Opening data with Fiji (ImageJ)**	
1.1	This pipeline is optimized for 3D images (stacked images). Most file formats can be opened directly from the File menu: FILE > OPEN… or FILE > IMPORT > [TYPE OF YOUR FILES] Other formats might require new plugins or a manual input (e.g. dimensions for raw data files).	
2	**ROI mask creation**	
2.1	In this example dataset, particle-like objects of interest are located in a specific region of the brain (hippocampus). To extract this region, a mask is first drawn with the Segmentation Editor: PLUGINS > SEGMENTATION > SEGMENTATION EDITOR Two windows open: one interface with the view of the original image, one for the mask (black). To navigate in the stack, use the mouse wheel, the top scrollbar or the arrow keys; on a slice, use the 'Zoom' buttons and the hand grip (from the main ImageJ shortcuts) to be more precise.	
2.2	Choose a tool (line, polygon, circle or rectangle) from the main menu to draw a region of interest (ROI). You can use the circle brush by double clicking the circle selection tool — it proved very efficient.	
2.3	Once you are done on a slice, move to another slice with the mouse wheel, the top scrollbar or the arrow keys; repeat the process 20 slices further. Use the interpolate function — the first 'Tools' button — to fill the selection between a few slices.	
2.4	Then, add the selected pixels to the label 'Exterior': select it in the 'Labels' list and click the "+" button under 'Selection'. If another structure that corresponds to the same label needs to be segmented, repeat the process.	
2.5	The mask is created once you click 'OK'. Its filename ends with 'labels'.	

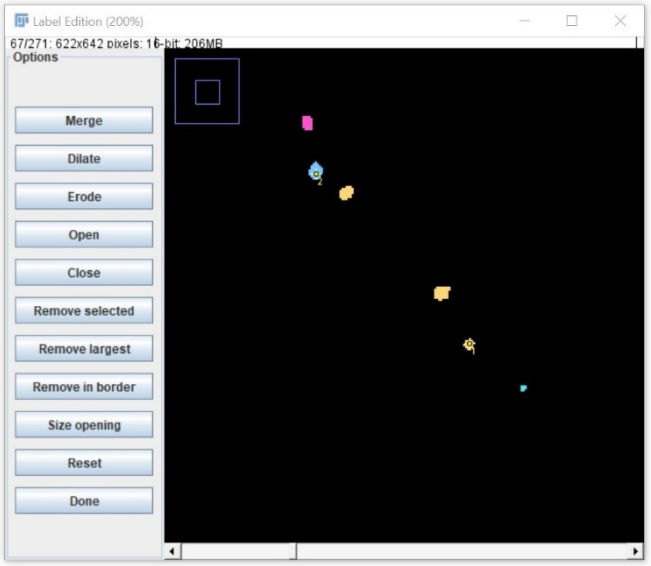
#	Step / Actions	Illustrations
3	**Application of the mask to extract hippocampus**	
3.1	To apply the mask to the image, you need to use this tool: PROCESS > IMAGE CALCULATOR… Pick the original image in the first input, and the newly created label image in the third input, with the 'Multiply' operation; check the first box only; confirm.	
3.2	After the computation, contrast needs to be adjusted, due to the conversion to 32-bit float: IMAGE > ADJUST > BRIGHTNESS/CONTRAST… You can save the result as a new image to begin with. Saving this file as a TIFF image is the most convenient: FILE > SAVE AS > TIFF…	
4	**Trainable WEKA Segmentation 3D**	
4.1	Depending on the computation capacity (RAM and CPU power), you should train using only a portion of the image: • You can crop the image around the ROI; save the ROI position through the ROI manager for reproducibility. • You could rather extract a few slices around the ROI: IMAGE > STACKS > TOOLS > MAKE SUBSTACK… At this point, you will use Fiji (ImageJ) to train and use the classifiers of the WEKA software, through the plugin: PLUGINS > SEGMENTATION > TRAINABLE WEKA SEGMENTATION 3D Two windows will pop up: a log window and the main window.	

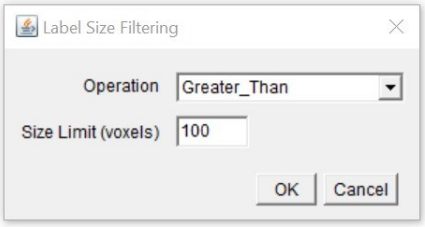
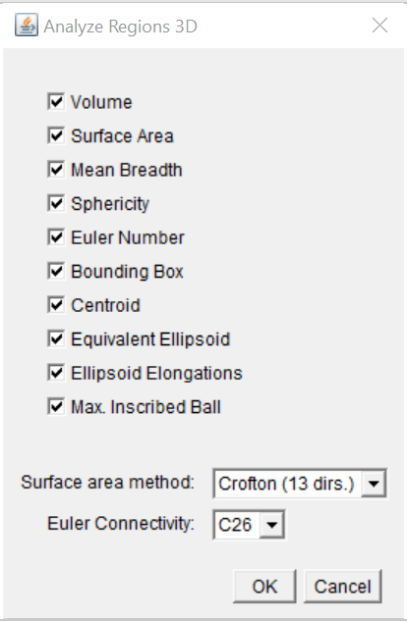
#	Step / Actions	Illustrations
4.2	To navigate in the stack, use the mouse wheel, the bottom scrollbar or the arrow keys; on a slice, use the 'Zoom' button (from the main ImageJ shortcuts) or the mouse wheel along with the `Ctrl` key to be more precise. Now, select samples of the image: • Select some of the objects of interest (you can use the brush) and click on "Add to class 1" under 'Labels'; • Select small areas of the background and click on "Add to class 2" under 'Labels'. You should do this on a few consecutive slices of the stack, especially if the objects of interest span over multiple slices.	
4.3	Pick complementary metrics to help the detection of the objects: add relevant ones in "Settings" (by default, only 'Mean' and 'Variance' are checked). Optional ImageScience plugin is needed for some of them. Once you have labelled multiple objects, you can train the algorithm by clicking on "Train classifier". An overlay appears with the result. Hide/show it while you browse the slices with "Toggle overlay". If you find mistakes in the segmentation, add some of the wrong pixels to the correct class and repeat the training (click again on "Train classifier"), until you are satisfied with the result. Then, click on "Get probability" and save the training with "Save classifier" and "Save data": they generate '*.model' and '*.arff' files respectively.	
4.4	Note that at this point, you used only a few slices to perform the training. To process the entire image, click on "Apply classifier" and select the image you saved at the step **3.2**. When a prompt appears, choose 'Yes' to generate probability maps. The operation may take some time depending on the size of the image.	
4.5	If at any point beyond, you are not satisfied with the results, you may have to go back to the training. At first, try to edit the training you performed on a few slices. If this fails to improve your results, you may have to use different slices.	

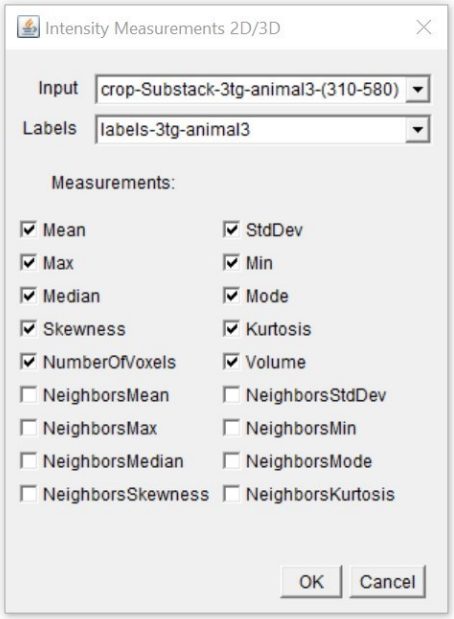
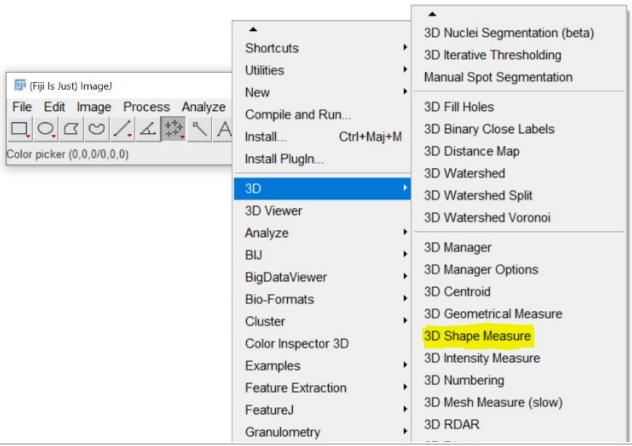
#	Step / Actions	Illustrations
	To do so, repeat step **4.1** and choose "Load data" (pick the '*.arff' file); you will have no visual feedback in the 'Labels', but the previous training dataset is loaded. Then, start a training to have something to begin with, and adjust labels as you did in step **4.3**. You can then replace the '.model' and '.arff' files with improved versions.	
5	**Thresholding**	
5.1	The new window that opens contains a 4-dimension image: the 4th dimension corresponds to your classes. You need only the layer of the foreground class (class 1), so use: IMAGE > STACKS > TOOLS > MAKE SUBSTACK… Set the 'Channels' range to the value "1" and click on "OK". This 3D probability map contains for each voxel the probability (range: $[0,1]$) of belonging to our foreground / object-of-interest class.	
5.2	Now, a threshold can be applied to the probability map: IMAGE > ADJUST > THRESHOLD… (shortcut: `Ctrl+Shift+T`) The regions that are to be segmented will appear in red to help you visualize; the percentage of the whole image it represents is displayed under the histogram. Use the two sliders to set the adequate range. In this case, the particle-like objects are randomly scattered. A high-quality training will result in a range close to $[0.8,1]$ for your objects of interest. Click on 'Apply'. A prompt appears, click on 'Convert to Mask'.	

#	Step / Actions	Illustrations
5.3	At the next window, uncheck 'Calculate threshold for each image'. Keep the default values for the other fields. Click on 'OK'.	
5.4	Now, a mask of the objects of interest has been obtained. Keep notes on the values used for the threshold, for example in the file name. You can save the new mask: FILE > SAVE AS > TIFF…	
6	**Labellisation**	
6.1	The goal of this step is to get distinct objects, instead of a binary image. MorphoLibJ provides such tool: PLUGINS > MORPHOLIBJ > BINARY IMAGES > CONNECTED COMPONENT LABELLING In the options, pick "26" for the 'Connectivity' and "16 bits" for the 'Type of result'. The label map gets an automatic name which ends in "-lbl".	
6.2	To visualize the results, you should change the colormap: PLUGINS > MORPHOLIBJ > LABEL IMAGES > SET LABEL MAP Default values are fine, but change them as you please.	

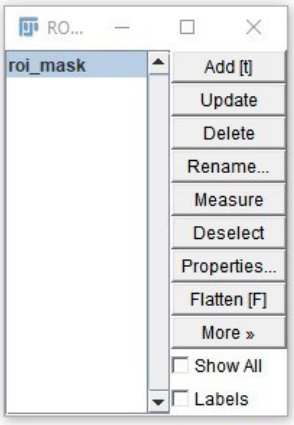
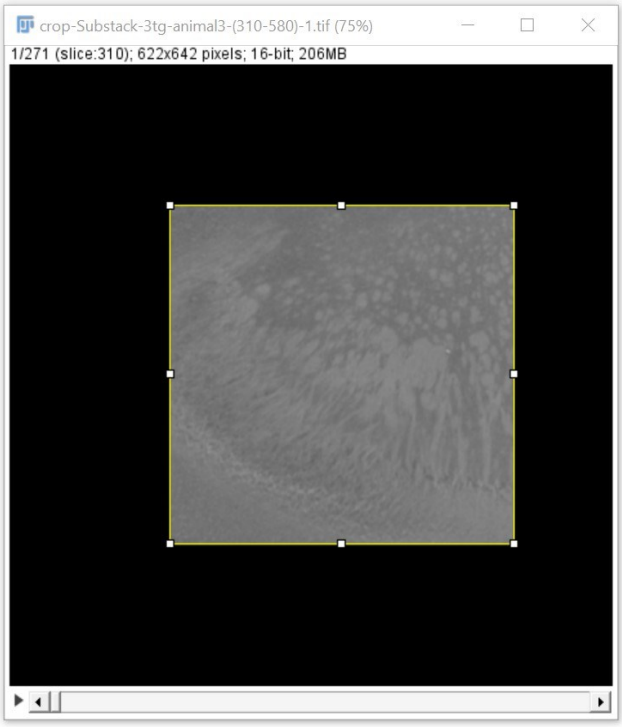
#	Step / Actions	Illustrations

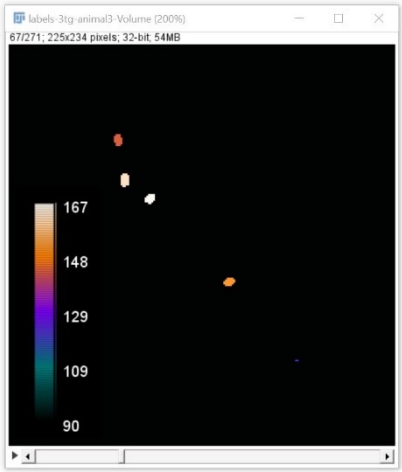
6.3	The label editor from MorphoLibJ can be used to remove abnormal labels objects:
	PLUGINS > MORPHOLIBJ > LABEL IMAGES > LABEL EDITION
	
	• Click on labels to select them;
	• Use "Merge" on selected elements that should be part of the same label;
	• Apply morphological operations to all labels:
	  ○ "Dilate" / "Erode" (add / remove pixels on the edges of the labels),
	  ○ "Close" (fill the holes in a label),
	  ○ "Open";
	• Other options are self-explanatory.
	• You can reset at any time: try the different options.
	
	Once you have cleaned the slices, click on 'Done'. The new label map has an automatic name which ends in "-edited".
		

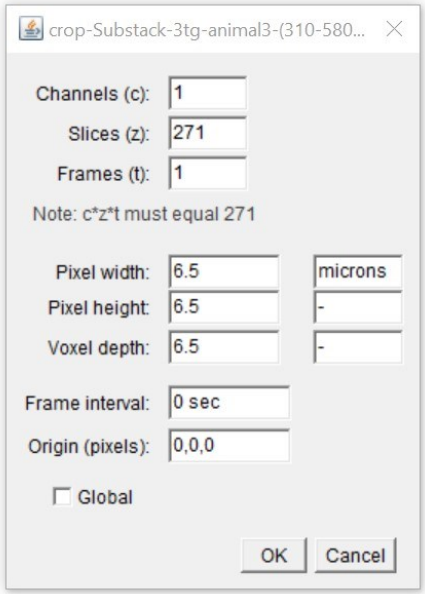
6.4	MorphoLibJ provides other tools for label cleaning, such as filtering according to size: PLUGINS > MORPHOLIBJ > LABEL IMAGES > LABEL SIZE FILTERING Depending on what you know of the objects of interest, you can define a range of sizes to <u>keep</u>, by using this tool twice (upper bound with "Lower_Than" and lower bound with "Greater_Than"). The tool appends "-sizeFilt" at each run.	
7	**Analysis with MorphoLibJ and 3D ImageJ Suite**	
7.1	Now, various metrics about the labels can be generated with MorphoLibJ: PLUGINS > MORPHOLIBJ > ANALYZE > ANALYZE REGIONS 3D The default settings enable all the metrics. Keep 'Surface area method' and 'Euler Connectivity' to the values shown on the right. Click on 'OK'. A table will open with the characteristics for each labeled object: save it before continuing.	

#	Step / Actions	Illustrations
7.2	We will repeat the process with another tool to get grayscale metrics: PLUGINS > MORPHOLIBJ > ANALYZE > INTENSITY MEASUREMENTS 2D/3D Set the 'Input' field to the image from step **3.2** (full stack) or **4.1** (substack) and the 'Label' field to the label map you generated. The default settings enable all the metrics. 'Neighbors*' metrics give information about connected labels, but we will not have any (the step **6.1** merged the connected components): disable these metrics. Click on 'OK'. A table will open with the characteristics for each labeled object: save it before continuing.	
7.3	The 3D ImageJ Suite provides the same type of measures, among others. Note that the discrete estimation of compactness and sphericity from the 3D ImageJ Suite may be less variable for small objects. Compute them with: PLUGINS > 3D > 3D SHAPE MEASURE Close the "Results" spreadsheet after saving it, as new data is appended to it.	

Extras

#	Step / Actions	Illustrations
A	**Optimal management of ROIs**	
A.1	To speed up resource-intensive computations, it is advised to crop the images around regions of interest. To be able to reproduce results, traces should be kept of which steps you did. ANALYZE > TOOLS > ROI MANAGER… With the ROI Manager, you can easily keep (`Ctrl + T`) the ROIs, adjust their size and positions, and save them separately (recommended) or as a whole. Then, open the '*.roi' file to get them back instantly.	
A.2	In case you need to reverse a cropping edition, for one slice only: just copy-paste the cropped slice into the currently selected ROI of the original image. In case you need a black background around the restored ROI, multiply the target image by zero with: PROCESS > MATH > MULTIPLY…	
A.3	In case you need to reverse a cropping edition, for a full stack: you'll need the top-left coordinate of the ROI. Go to the ROI Manager and right-click on "Specify…": note the X and Y coordinates. Use then the stack inserter: IMAGE > STACKS > TOOLS > INSERT…	

#	Step / Actions	Illustrations
B	**Management of image borders**	
B.1	*Labels* — Labels on the borders of the images, along the 3 dimensions, can be responsible of outliers in a statistical study of extracted metrics. MorphoLibJ implements a way to suppress them: PLUGINS > MORPHOLIBJ > LABEL IMAGES > REMOVE BORDER LABELS	
B.2	*ROIs* — To counter the same problem at an early stage, it can be useful to enlarge the selection, and thus the ROI: EDIT > SELECTION > ENLARGE… Do not forget to "Update" the ROI in the ROI Manager.	
B.3	*Extend image* — You can also enlarge the image by adding zero-value pixels on the sides, with: IMAGE > ADJUST > CANVAS SIZE… However, it requires to add manually empty slices on the Z axis. Duplicate a few slices and set them to zero: IMAGE > DUPLICATE… PROCESS > MATH > MULTIPLY… (with value "0.0") Then, add these slices at the beginning and at the end of the image (as shown on the right): IMAGE > STACKS > TOOLS > CONCATENATE…	
C	**Manipulation of labels with MorphoLibJ**	
C.1	You can apply morphological filters. For example, you can fill holes in binary images or labelled images: PLUGINS > MORPHOLIBJ > FILL HOLES (BINARY/GRAY) or PLUGINS > MORPHOLIBJ > MORPHOLOGICAL FILTERS (3D)	
C.2	Measure-related colormaps can be built. Hence you can represent any measure from any spreadsheet, using: PLUGINS > MORPHOLIBJ > LABEL IMAGES > ASSIGN MEASURE TO LABEL Then you can setup a LUT related with these values in the LUT editor: IMAGE > COLOR > EDIT LUT… Finally, add a colormap legend with: ANALYZE > TOOLS > CALIBRATION BAR…	

#	Step / Actions	Illustrations
D	**Scale for 3D measures in standard unit**	
D.1	Set the scale from the image properties: IMAGE > PROPERTIES…	
D.2	Set the scale from a known length. Trace a line and go to: ANALYZE > SET SCALE… If you know the voxel resolution, you do not need to trace a line.