MORPHIOUS: an unsupervised machine learning workflow to detect the activation of microglia and astrocytes – a Tutorial

Joey Silburt

This readme manual is an accompaniment to the accompanying MORPHIOUS research paper, available at: (Silburt & Aubert, 2022)

An accompanying video tutorial will soon be made available:

Table of Contents

Brief overview:

MORPHIOUS uses a one-class support vector machine to detect clusters of activated microglia or astrocytes in tissue sections after referencing examples of normal, untreated tissue. MORPHIOUS requires a discrete training set of control images where cells of interest are known to be at baseline or resting states. After training, MORPHIOUS can be applied to a second test set where it can identify the presence of clusters of activated cells.

Step 1: Guidelines for preparing your images for MORPHIOUS.

Preparing your immunofluorescence sample:

MORPHIOUS works by learning a definition of "normal" from a set of control samples, and inferring clusters of activation from this definition. As such, it is critical to train MORPHIOUS with a sufficient distribution of what normal tissue looks like. How that is defined is up to the user. In general, a more narrow definition of "normal", might therefore produce better results. For example, in the original paper, MORPHIOUS was trained on tiled images of entire hippocampal regions – a reasonably homogenous baseline distribution of cells (Silburt & Aubert, 2022). Nevertheless, MORPHIOUS still proved to be quite robust when trained on whole brain regions (Kofoed et al., 2021).

While there is no definitive rule, it is recommended users follow standard immunohistochemical sampling practices and select, at a minimum, multiple samples (i.e., 3+) and multiple fields of view per sample (i.e., 3+). As with all immunohistochemical analysis approaches, choosing biased fields of view will bias the results (Jensen, 2013; Meyerholz & Beck, 2018; O'Hurley et al., 2014). Consequently user's should use representative samples when training and testing MORPHIOUS.

In general, the success of any immunofluorescence analysis depends on the consistency of the immunofluorescence stain. In my experience, this means that as much as possible, samples should be stained at the same time using the same buffers and solutions. Failure to do this might produce samples with significant difference in immunofluorescence brightness which can impact the capacity to perform subsequent analysis. In general, samples should be imaged using the same settings so that they can be faithfully compared.

Preparing your images for MORPHIOUS:

The MORPHIOUS macros are built with the capacity to perform standard preprocessing steps (e.g., background subtraction, image contrasts, despeckling). Consequently, these steps do not need to be performed before hand. However, if it's desired, users can simply turn off these preprocessing steps within the MORPHIOUS macros, and preprocess the images themselves before using MORPHIOUS. Images should be in **.tif** format.

Delineating a region of interest

It is not necessary to predefine a region of interest. However, if large tiled images are being analyzed, the analysis can be focused on a specific region by using the polygon tool in ImageJ to delineate a region of interest (Fig 1) and clearing the outside (Edit -> Clear outside). Delineating a region of interest can further be used to remove artifacts which can sometimes occur on the edge of samples when staining. User's should show extreme caution when cropping images, and line with best principles and practices, ensure that they are fairly representing the sample (Jensen, 2013; Meyerholz & Beck, 2018; O'Hurley et al., 2014). It should be noted that seemingly black spaces in a tiled image can sometimes produce spurious skeletons (Fig 1B) which can impact results. However, this may only present as a problem for images with large regions of blank space.

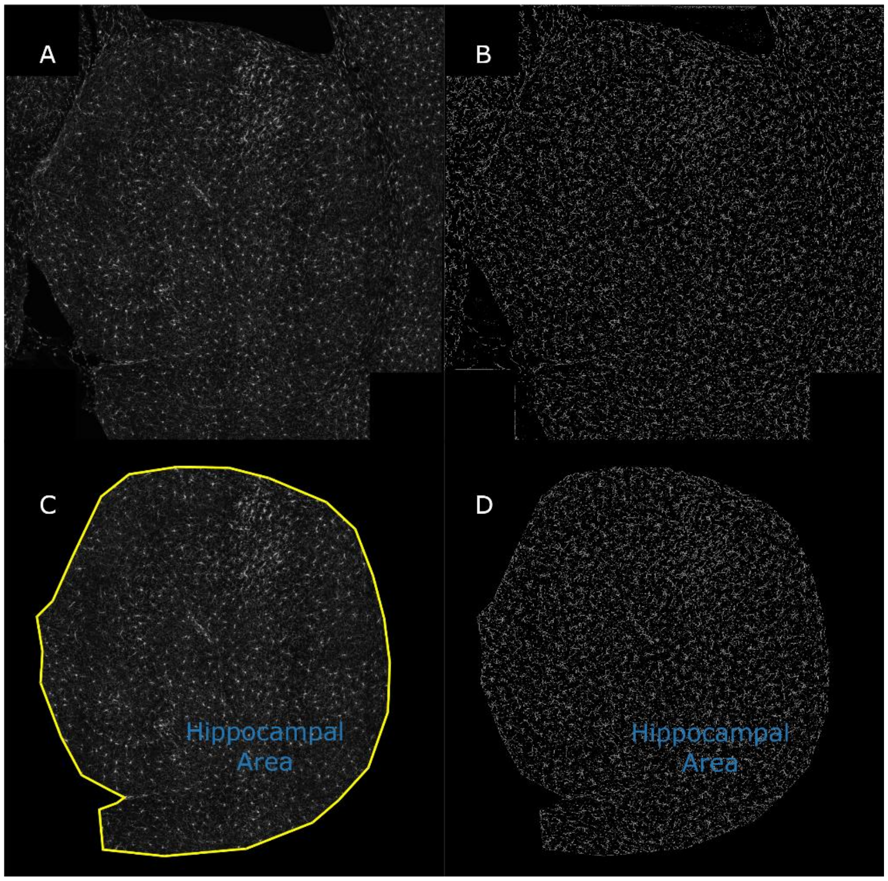


Fig 1: Images with uncropped edges (A) can sometimes produce spurious skeleton traces (B). Fairly delineating a region of interest and clearing the outside (C) can eliminate such artifacts (D).

Step 2: Extract features from your images using the provided imageJ macros

Four feature extraction macros are provided to extract a variety of features.

1) get_intensity_measures.ijm

   Evaluates various intensity measures for the image including “Mean”, “IntDen”, “circularity”, etc.

2) get_fractal_measures.ijm

   Evaluates the fractal dimension.

3) get_branch_features.ijm

   Evaluates various skeletal metrics, such as branch length, number of branches, number of junctions, etc.

4) count_microglia.ijm

Identifies the soma body of Iba1$^+$ microglia. Regions of interest corresponding to these somas are saved. Moreover, soma features, such as size and circularity, etc., are evaluated.

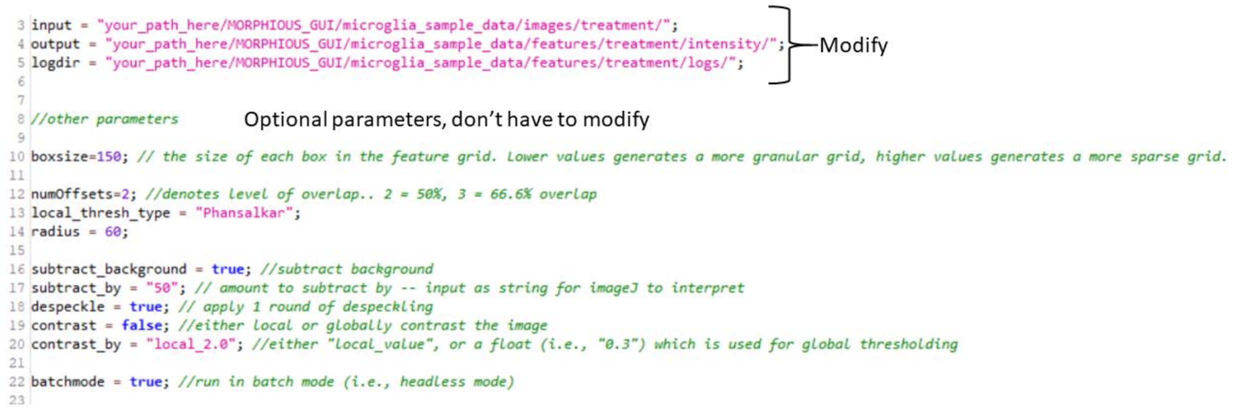
5) count_astrocytes.ijm

Identifies the soma body of S100β+ astrocytes. Regions of interest corresponding to these somas are saved. Moreover, soma features, such as size and circularity, etc., are evaluated.

In general, to use these macros, you are only required to input the location of your images, and output directories (Fig. 2). These should be created in advance. Control and test image datasets should be separated into separate directories. Moreover, extracted features for control and test images should be saved in separate directories. All macros also require the user input the location of a directory for a log file, where all parameters are saved. The get_fractal_measures.ijm, get_branch_features.ijm, and count_microglia.ijm, and count_astrocytes, ask for the input of additional directories for saving binarized images.

Please note: directories should be separated by a forward slash, i.e., "/" not a "\", which is used as an escape code character in ImageJ's macro language.

For example, to evaluate intensity measures for the control dataset provided in the sample data, an input scheme would look like this:



```
input = "your_path_here/MORPHIOUS_GUI/microglia_sample_data/images/treatment/";
output = "your_path_here/MORPHIOUS_GUI/microglia_sample_data/features/treatment/intensity/";
logdir = "your_path_here/MORPHIOUS_GUI/microglia_sample_data/features/treatment/logs/";


//other parameters

boxsize=150; // the size of each box in the feature grid. Lower values generates a more granular grid, higher values generates a more sparse grid.

numOffsets=2; //denotes level of overlap.. 2 = 50%, 3 = 66.6% overlap
local_thresh_type = "Phansalkar";
radius = 60;

subtract_background = true; //subtract background
subtract_by = "50"; // amount to subtract by -- input as string for imageJ to interpret
despeckle = true; // apply 1 round of despeckling
contrast = false; //either local or globally contrast the image
contrast_by = "local_2.0"; //either "local_value", or a float (i.e., "0.3") which is used for global thresholding

batchmode = true; //run in batch mode (i.e., headless mode)
```

For get_intensity_measures.ijm, get_fractal_measures.ijm, and get_branch_features.ijm a number of additional parameters can be altered at the users the discretion. These include:

Boxsize: The size of the box created in the grid. Smaller values will generate a more granular feature map, while larger values will generate a more coarse feature map. The boxsize should be sufficiently large to capture meaningful cellular features (e.g., branching complexity). Moreover, boxsizes which are too small may significantly increase processing time.

numOffsets: is inversely proportional to the degree a box is slid in the x and y directions. A numOffsets of 2 means that the box is translated by 50% in the x and y directions. A numOffsets value of 3 would result in the box being translated by 33% in the x and y directions. Larger values of numOffsets results in more granular feature maps.

Local_thresh_type: The autolocalthreshold type used for imageJ, it is recommended to not change this value.

Radius: corresponds to the radius parameter in ImageJ autolocalthreshold.

Subtract_background: whether to apply a rolling box background subtraction.

Subtract_by: The radius parameter for the imageJ background subtraction method.

Despeckle: applies one round of image despeckling.

Contrast: whether to apply either a global or local contrast to the image. If false, the image will not be contrasted regardless of the value of "contrast_by".

Contrast_by: defines the contrast method and the degree of contrasting. Values of the form, "local_x.x" will perform local thresholding with the x.x referring to the maximum parameter value for ImageJ's Enhance Local Contrast (CLAHE) method. If only a value is provided e.g., "0.3", a global contrast is applied, with the value provided used as the percent saturation parameter.

Batchmode: Whether to run in ImageJ's batch mode (i.e., headless). Setting Batchmode to true will make the extraction run faster.

The macros, count_astrocytes.ijm, and count_microglia.ijm, contain addition parameters which the user is discouraged from modifying, as these have been experimentally optimized to improve soma detection. Nevertheless, it is possible that other parameters will yield better results depending on their specific imaging parameters.

Step 3: Open MORPHIOUS with python.

Ensure python is installed on your machine. Downloading python via the Anaconda package manager is recommended, as it comes with several important packages pre-installed.

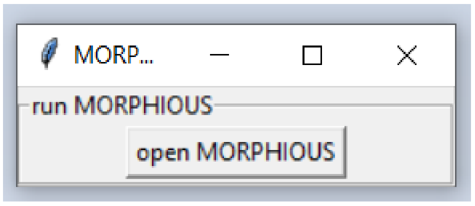
If not already installed, MORPHIOUS depends on the numpy, scipy, pandas, and sklearn libraries.
To install this with python, simply type:

```
pip install numpy

pip install scipy

pip install pandas

pip install sklearn
```

Using a computer terminal, locate the source_code directory. To accomplish this, users are encouraged to lookup how to navigate file directories via the command line. In general this can be accomplished via the "cd" command. User's can list the files in a current directory using the "dir" command (for windows) or the "ls" command for mac/linux. Once in the source_code directory, type:

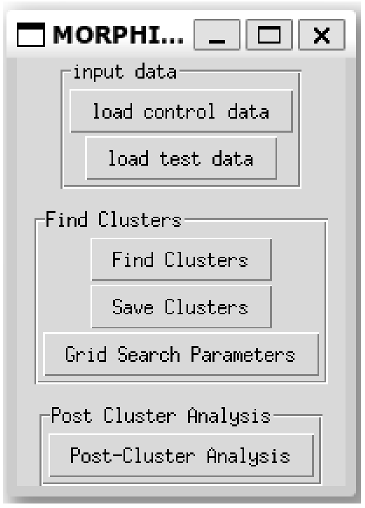
```
python morphious_main.py
```

This will produce the main MORPHIOUS frame:

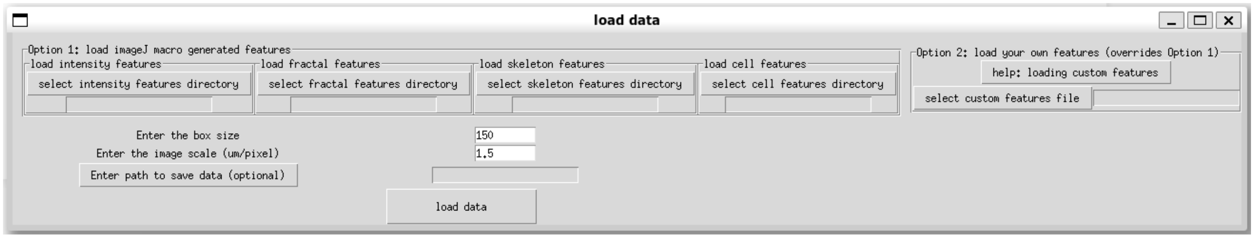


Click open MORPHIOUS to proceed, yielding the main options dialog.

Step 4: Load training and test sets into the program:



For a typical work flow, the user should load both control and test data sets. Selecting either the "load control data" or "load test data" buttons will yield a panel prompting the user to locate the outputs of the imageJ macros from step 2.


load data
Option 1: load imageJ macro generated features
load intensity features
select intensity features directory
load fractal features
select fractal features directory
load skeleton features
select skeleton features directory
load cell features
select cell features directory
Option 2: load your own features (overrides Option 1)
help: loading custom features
select custom features file
Enter path to save data (optional)
load data

Enter the box size: 150
Enter the image scale (um/pixel): 1.5

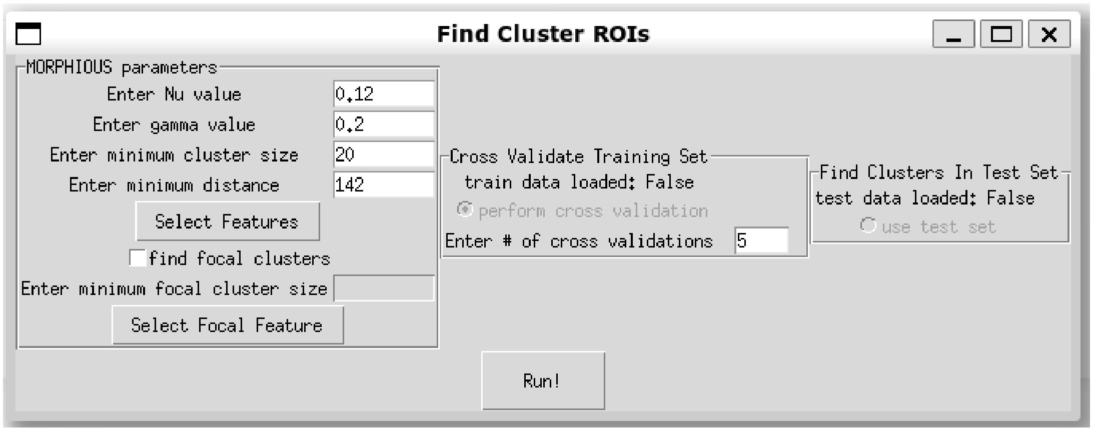
The user is not required to load all of intensity features, fractal features, skeleton features, and cell features, however, data from atleast one of these sources is required.

The user is also prompted to enter the boxsize that was chosen during the feature extraction phase in the imageJ macros. Finally, the user should input the scale of the image. This can be found in imageJ via Analyze → set scale.. on an open image. For convenience the image scale is also saved in the logfile for each of the intensity, fractal, and skeleton feature extraction macros.

The merged feature data frame can be saved by entering a path to save. This file can in turn be reloaded into MORPHIOUS via option 2. Alternatively, the user can load their own collected features via option 2. This dialog expects a single .csv file. At a minimum, this file must have a column called "file" which delineates between individual sample, and the coordinates "BX" and "BY" which MORPHIOUS uses to identify spatial clusters.

Step 5: Select MORPHIOUS Parameters

After data is loaded, users can proceed to the "Find clusters" frame.


Find Cluster ROIs

MORPHIOUS parameters

Enter Nu value	0.12
Enter gamma value	0.2
Enter minimum cluster size	20
Enter minimum distance	142

Select Features

find focal clusters

Enter minimum focal cluster size

Select Focal Feature

Cross Validate Training Set

train data loaded: False

perform cross validation

Enter # of cross validations 5

Find Clusters In Test Set

test data loaded: False

use test set

Run!

The user has several parameters they need to set, however, in general, these parameters are determined experimentally via a grid search (see below).

Nu: This parameter governs the 'misclassification rate' of the one-class support vector machine. Larger values will yield a greater number of outliers, and therein a greater degree of clustering. Depending on your experimental setup, optimal values may fall somewhere between 0.01 and 0.2.

Gamma: This parameter governs the complexity of the radial basis function kernel for the ocSVM. Values should generally fall between 0.1 and 0.25; In our experience, it is not advisable for values to exceed 0.3, as this leads to poor performance.

Minimum cluster size: This parameters controls the number of outliers required to be within the minimum distance of a neighbouring outliers to be labelled a cluster. For default feature extraction parameters, this value may fall between 18 and 22. However, reducing the box size, or increasing the number of box offsets will increase the minimum cluster size.

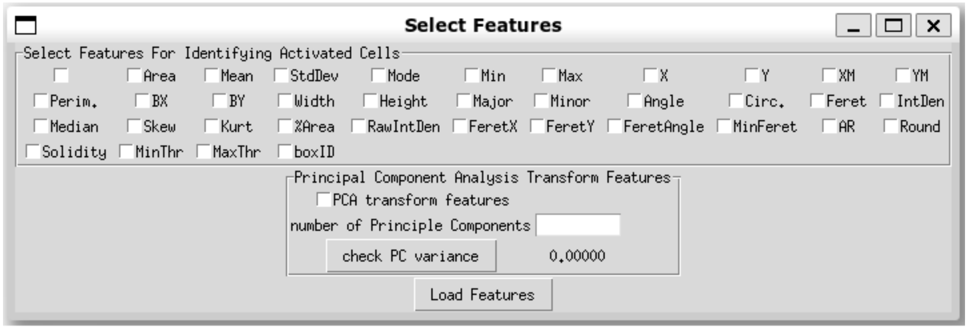
Enter minimum distance: The distance between neighbouring clusters to determine if they are part of a cluster. In our experience, a good default setting is to set this distance as the size of the hypotenuse of a boxsize by boxsize triangle.

Find focal clusters: if selected, MORPHIOUS will evaluate the presence of focal clusters. Focal clusters represent the most extreme variants of proximal clusters, based upon integrated density values.

Minimum focal cluster size: See minimum cluster size, in our experience, a good default setting is 5, but this can be as low as 1 depending on results.

Before proceeding to clustering, the user should select the features they wish to identify activated clusters with via the “Select Features” button. If user’s also want to identify focal clusters, they can change the focal feature to use via the “Select Focal Feature” button.

Step 6: Select Features.


Select Features

Select Features For Identifying Activated Cells

☐	☐ Area	☐ Mean	☐ StdDev	☐ Mode	☐ Min	☐ Max	☐ X	☐ Y	☐ XM	☐ YM
☐ Perim.	☐ BX	☐ BY	☐ Width	☐ Height	☐ Major	☐ Minor	☐ Angle	☐ Circ.	☐ Feret	☐ IntDen
☐ Median	☐ Skew	☐ Kurt	☐ %Area	☐ RawIntDen	☐ FeretX	☐ FeretY	☐ FeretAngle	☐ MinFeret	☐ AR	☐ Round
☐ Solidity	☐ MinThr	☐ MaxThr	☐ boxID							

Principal Component Analysis Transform Features

☐ PCA transform features

number of Principle Components

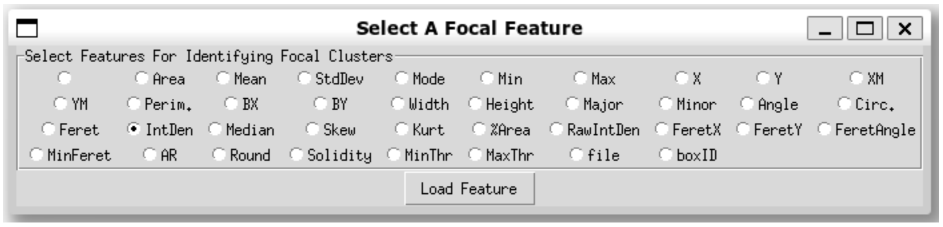
check PC variance 0.00000

Load Features

Features extracted via the ImageJ feature extraction macros are available for selection here. The user also has the option to transform the features via principle component analysis. Please be aware, some features available via this dialogue, such as BX/BY contain no meaningful information about the difference between an activated and non-activated cell, and so should be avoided. For more information on the meaning of each feature, read the imageJ documentation.

PCA transformation: Our recommendation is to PCA transform features with enough principle components to retain >99% of the feature variance. A button is available to assess the current variance encapsulated by the selected number of principle components.

Selecting a focal feature:


Select A Focal Feature

Select Features For Identifying Focal Clusters

○	○ Area	○ Mean	○ StdDev	○ Mode	○ Min	○ Max	○ X	○ Y	○ XM
○ YM	○ Perim.	○ BX	○ BY	○ Width	○ Height	○ Major	○ Minor	○ Angle	○ Circ.
○ Feret	[✓ IntDen]	○ Median	○ Skew	○ Kurt	○ %Area	○ RawIntDen	○ FeretX	○ FeretY	○ FeretAngle
○ MinFeret	○ AR	○ Round	○ Solidity	○ MinThr	○ MaxThr	○ file	○ boxID		

The “select a focal feature dialog” is similar to the select features window, and allows a user to change the default focal feature. The default variable, “IntDen” is a good choice for most cases.

Step 7: using cross validation in the training set and finding clusters in the test set.

MORPHIOUS works by learning a set of parameters which result in no activation clusters in control tissue, while maximizing the set of clustering in the test tissue. Cross-validation is used to evaluate whether that selected MORPHIOUS parameters yield clusters of activated cells in the control tissues.

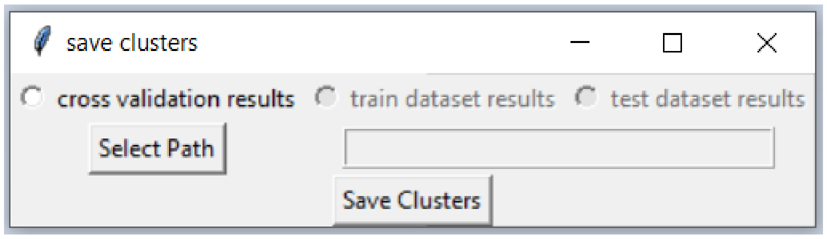
With cross-validation, the training set is randomly divided among an equal number of subsets (as determined by the user). For each round of cross-validation, one of the subsets is used as the test set, while the remainder are used to train the classifier. This is repeated until all data subsets have been evaluated as the test set. In this way, each control set image can be evaluated for the presence of spurious activation clusters. Detecting no clusters in the cross-validation step is needed to validate the integrity of predicted clusters in the test set. Cross-validation only ever searches for “proximal” clusters, therefore selecting the “find focal clusters” option does not impact the cross-validation results.

Performing a cross validation is designated by selecting the “perform cross validation” radio button. The number of cross-validation subsets is indicated by “the # of cross validations” parameter. Increasing the number of subsets will improve the model performance, however, may significantly increase computational time. The number of selected cross-validations cannot exceed the number of images loaded in the training set.

After a set of MORPHIOUS parameters has been determined, clusters can be detected in the test set. This is accomplished by selecting the “use test set” feature. Both the training set and the test set are needed to use the test set.

Step 8: Save output cluster files.

After finding clusters, the outputs of these can be saved.



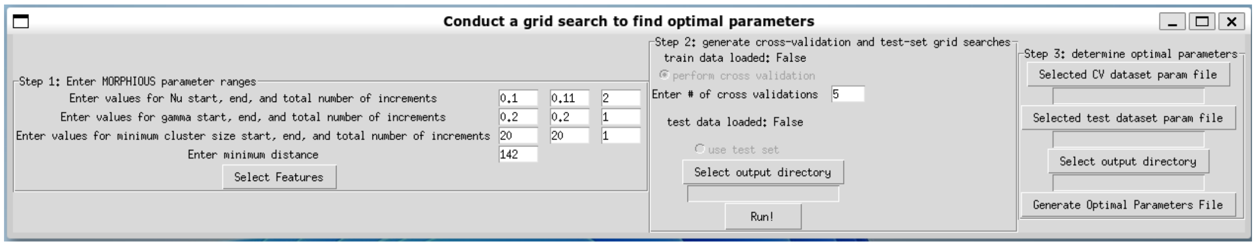
Saving results will generate "proximal" clusters and "unclustered" directory outputs. In addition, if the "find focal clusters" option was selected, "focal" and "combined" clusters will also be outputted. Combined clusters represent the combination of both proximal and focal clusters. Whereas proximal clusters likely contain focal clusters as well, "proximal only" clusters can be evaluated by subtracting combined clusters from focal clusters. For each individual image, the spatial coordinates of each cluster box is saved. Files in the unclustered directory are blank, but serve as an index to indicate which images had no clustering.

The "make_cluster_ROIs.ijm" macro can be used to generate imageJ region of interests (ROIs) from the generated cluster files (see step 10).

Step 9: Using a grid search to find optimal MORPHIOUS parameters

In most cases a grid search will needed to be conducted to find optimal MORPHIOUS parameters. Briefly, a grid search iterates through a range of possible parameters to identify optimal parameter values.

To conduct a grid search, click the "Grid Search Parameters" button on the options frame.


Conduct a grid search to find optimal parameters

Step 1: Enter MORPHIOUS parameter ranges

Enter values for Nu start, end, and total number of increments	0.1	0.11	2
Enter values for gamma start, end, and total number of increments	0.2	0.2	1
Enter values for minimum cluster size start, end, and total number of increments	20	20	1
Enter minimum distance	142		

Step 2: generate cross-validation and test-set grid searches
train data loaded: False
[✓ perform cross validation]
Enter # of cross validations: 5
test data loaded: False
use test set

Step 3: determine optimal parameters

Select Features
Select output directory
Run!
Selected CV dataset param file
Selected test dataset param file
Select output directory
Generate Optimal Parameters File

For step 1, input values for the start, end, and total number of increments for then nu, gamma, and minimum cluster size parameters. The increment defines the number of equally spaced values between the start and end values. Both start and end values are inclusive. Therefore, a nu range of start: 0.1, end: 0.11, number of increments: 2, will yield 0.1, and 0.11 as values. A nu range of start: 0.1, end: 0.2, increments: 6, will yield 0.1, 0.12, 0.14, 0.16, 0.18, and 0.2, as values. For discussion on the minimum distance, refer to the "step 5: selecting MORPHIOUS parameters" section.

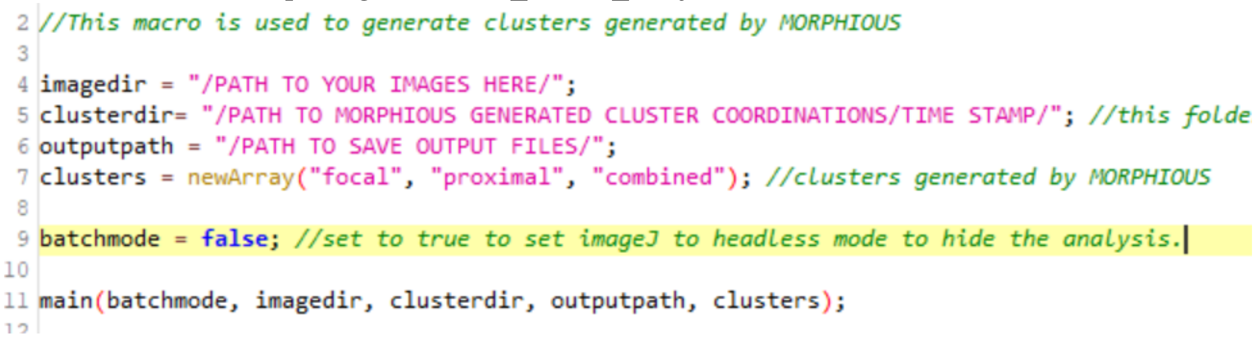
After features are selected, the user should proceed to step 2. Grid searches will be needed to be conducted separately for both the train dataset (via cross-validation), and the test dataset. Depending on the size of the dataset, a grid search can take a substantial amount of time. It is recommended to test the timing of a single "Find Clusters" workflow to help guide the breadth of grid search selected. Cross validation grid search result files are distinguished by "…X-fold_CV…" in the resultant file name.

Once both the training grid search and test grid search files are generated, proceed to step 3. In step 3, select the relevant files, select an output directory for the resultant summary file. Afterwards push the “generate optimal parameter file” button. The summary file will indicate all parameters which resulted in no clustering in the training dataset. Parameters will further be sorted in order of which parameters yielded the greatest degree of clustering in the test dataset. Thus, from this list, optimal parameters for nu, minN, and gamma, can be selected.

Step 10: Creating MORPHIOUS generated ROIs

After you have finished with generating the cluster coordinates, you can use the provided imageJ macro: “make_cluster_ROIs.ijm” to create the MORPHIOUS generated clusters.

First, let’s start with opening the “make_cluster_roi.ijm” macro.



```
//This macro is used to generate clusters generated by MORPHIOUS

imagedir = "/PATH TO YOUR IMAGES HERE/";
clusterdir= "/PATH TO MORPHIOUS GENERATED CLUSTER COORDINATIONS/TIME STAMP/"; //this folde
outputpath = "/PATH TO SAVE OUTPUT FILES/";
clusters = newArray("focal", "proximal", "combined"); //clusters generated by MORPHIOUS

batchmode = false; //set to true to set imageJ to headless mode to hide the analysis.

main(batchmode, imagedir, clusterdir, outputpath, clusters);
```

This macro has 5 parameters, similar to the feature extraction macros.

imagedir: The location of your images, for example, “D:/My_Test_Images_Location/”.

clusterdir: The location of the cluster coordinate files saved in step 7. Select the folder that looks like a timestamp. This folder will contain separate subdirectories pertaining to each cluster (e.g., focal, proximal, combined).

outputpath: The location of an output directory.

clusters: This array corresponds to the directories present within the selected “clusterdir”, the most likely values will be “focal”, “proximal”, “combined”, OR, simply “proximal”.

batchmode: if set to true, the macro will run faster and in headless mode, hiding the various image popups etc.

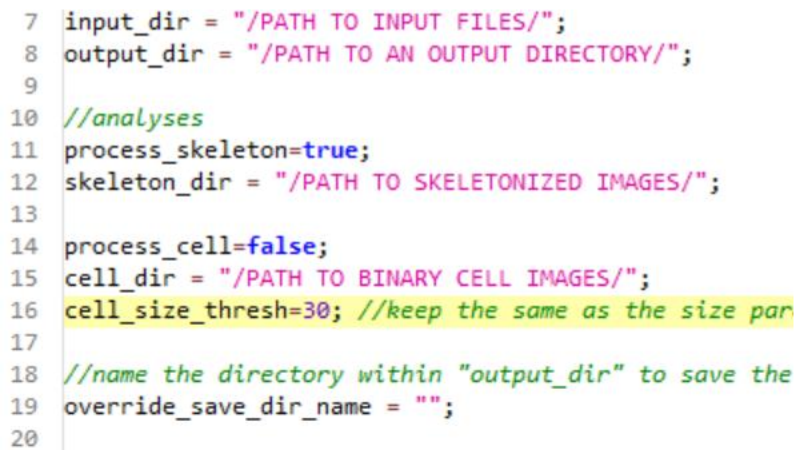
Step 11: post-MORPHIOUS analysis

The final step is to analyze the results, which can be accomplished using the “post_morphious_analysis.ijm” macro and the post-cluster analysis button. This macro can be used to analyze both MORPHIOUS clusters, and regular unclustered images.

Step 11.1: running the post-MORPHIOUS analysis macro

This macro has a number of settings that need to be filled in by the user:

1) Setting paths to image files and outputs:



```
input_dir = "/PATH TO INPUT FILES/";
output_dir = "/PATH TO AN OUTPUT DIRECTORY/";

//analyses
process_skeleton=true;
skeleton_dir = "/PATH TO SKELETONIZED IMAGES/";

process_cell=false;
cell_dir = "/PATH TO BINARY CELL IMAGES/";
cell_size_thresh=30; //keep the same as the size par

//name the directory within "output_dir" to save the
override_save_dir_name = "";

```

input_dir: The path to the original image files to be analyzed.

Output_dir: The path to an output directory to save the results to.

Process_skeleton: If true, skeleton analysis will be performed. This requires the user to set a path to the skeleton image files (i.e., such as those generated via the get_branch_features.ijm macro).

Skeleton_dir: the path to a directory containing the skeleton image files.

Process_cell: If true, skeleton analysis will be performed. This requires the user to set a path to the skeleton image files (i.e., such as those generated via the get_branch_features.ijm macro).

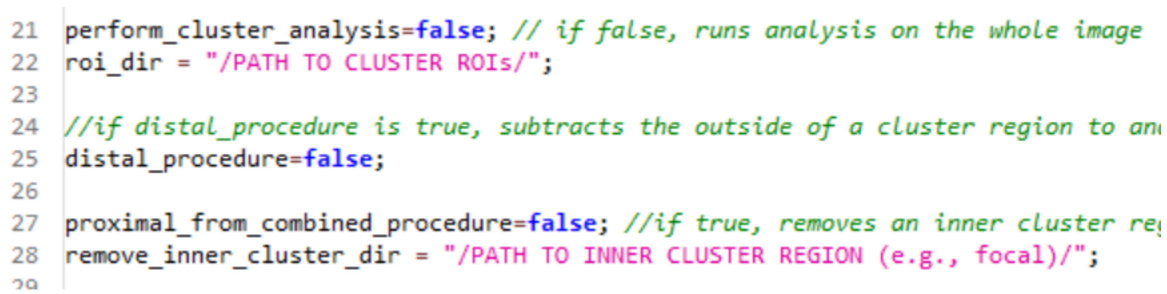
cell_dir: the path to a directory containing the binarized cell soma image files (i.e. produced via the count_microglia.ijm / count_astrocyte.ijm macros).

Cell_size_thresh: the threshold for the size of bone fide cell bodies. should be kept the same as the value used in the count_microglia.ijm / count_astrocyte.ijm macros

Override_save_dir_name: The name of the output folder (created by the macro in output_dir) for which the analysis data will be saved to. This value does not need to be set as the macro will auomatically generate meaningful default names. However, if the user chooses, they can set the name of the output folder by filling in this value.

2) Configuring the post-cluster analysis

After setting the necessary analyses and paths, the user must set several parameters for their cluster analysis.



```
perform_cluster_analysis=false; // if false, runs analysis on the whole image
roi_dir = "/PATH TO CLUSTER ROIs/";

//if distal_procedure is true, subtracts the outside of a cluster region to an
distal_procedure=false;

proximal_from_combined_procedure=false; //if true, removes an inner cluster re
remove_inner_cluster_dir = "/PATH TO INNER CLUSTER REGION (e.g., focal)/";
```

Perform_cluster_analysis: if set to true, the macro will use the MORPHIOUS identified clusters to perform analysis only on the clustered regions within the images. If left at false, the macro will perform it's analysis using the entire image provided, i.e., a whole-image analysis. This is used, for example, to analyze the control sample images which do not have clusters.

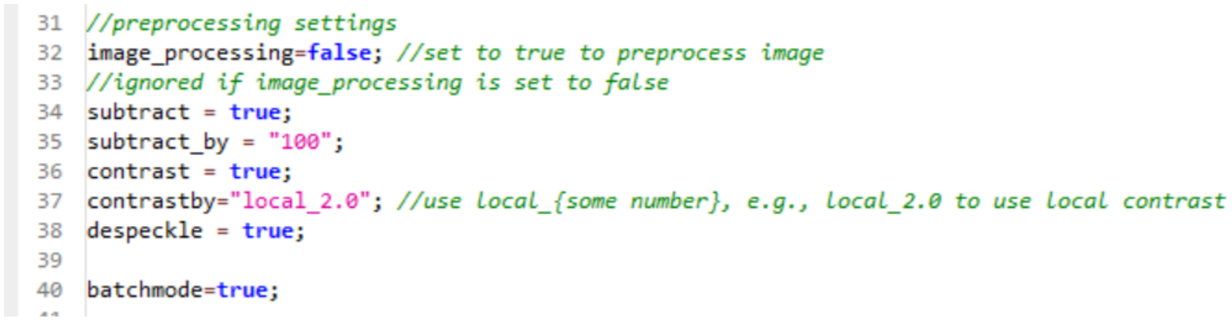
Roi_dir: The path to MORPHIOUS generated ROIs (I.e., the files generated from step 10), this value is ignored if **perform_cluster_analysis** is false.

Distal_procedure: The distal procedure will analyze the non-clustered areas in a treatment area. It will not run if perform_cluster_analysis is false. Consequently, the **roi_dir** should point to the cluster ROI that contains all clustered areas in an image. This will be either the proximal cluster ROIs (if focal clusters were not searched for), or combined cluster ROIs (i.e., proximal and focal clusters combined).

Proximal_from_combined_procedure: This procedure will remove an inner cluster region from an outer cluster region. Because proximal clusters typically contain focal clusters within them, focal regions must be removed in order to accurately represent the morphologies of proximal cells. Thus, this procedure will subtract the ROIs present within the **remove_inner_cluster_dir** folder from the **roi_dir** clusters. This procedure will not run if **distal_procedure** is set to true.

Remove_inner_cluster_dir: Path to inner clusters that will be removed from ROIs present within the **roi_dir.**

3) Applying image corrections

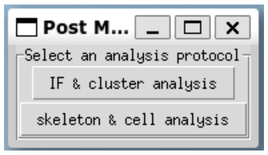


```
//preprocessing settings
image_processing=false; //set to true to preprocess image
//ignored if image_processing is set to false
subtract = true;
subtract_by = "100";
contrast = true;
contrastby="local_2.0"; //use local_{some number}, e.g., local_2.0 to use local contrast
despeckle = true;

batchmode=true;
```

These corrections are the same as in the feature extraction macros (see step 2).

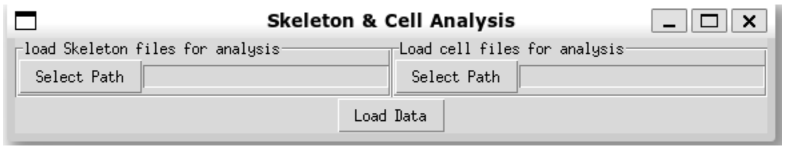
Step 11.2: Compiling the final results

Selecting the post-cluster analysis button on the main options frame will produce the following box:

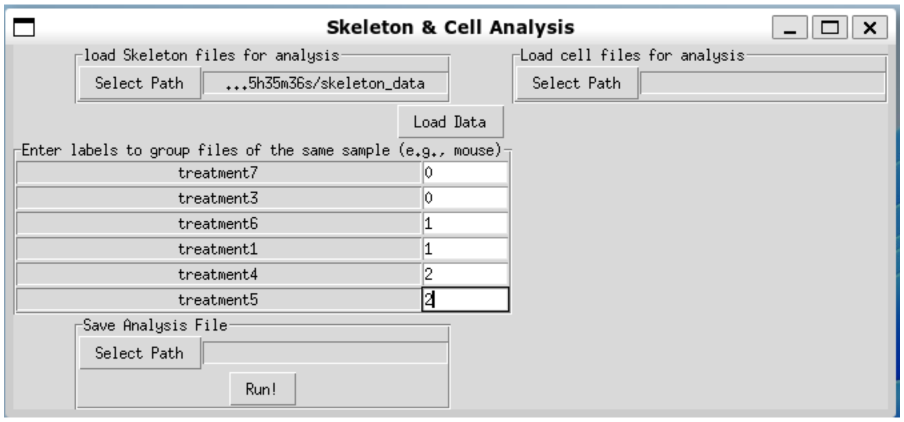


IF & cluster analysis processes the post-cluster analysis immunofluorescence files, while the skeleton & cell analysis processes the skeleton and cellular morphology files. Both buttons produce a similar analysis frame.

For example:


Skeleton & Cell Analysis
load Skeleton files for analysis
Select Path
Load cell files for analysis
Select Path
Load Data

Here the user can select the folder corresponding to their skeleton and or cell analysis files (i.e., the skeleton and cell directories produced in step 11.1). Please note, the user is not required to load both skeleton and cell analysis, but one must be chosen. After loading the data, the frame will transform to produce:



load Skeleton files for analysis: ...5h35m36s/skeleton_data

Enter labels to group files of the same sample (e.g., mouse)

treatment7	0
treatment3	0
treatment6	1
treatment1	1
treatment4	2
treatment5	2

Here the user has the opportunity to group the original image files into samples. In this example, the images “treatment7” and “treatment3” represent two images from the same mouse, which is denoted by both being labelled 0. Similarly, “treatment6” and “treatment1” come from the mouse, denoted by the label 1. Finally, “treatment4” and “treatment5” come from the same mouse, and so are grouped via the label 2. In the final analysis, these identified samples will be averaged together. This averaging is a weighted average which accounts for the size of each individual image. Finally, the user should select a path to save the results to and click run!