**ImageJ Script: CalculateAverageIntensity.py Documentation Version 1.0**

**Benjamin Lowe**

**27th February 2018**

# Installation of required software:

This script requires [Fiji](https://imagej.net/Fiji/Downloads) (ImageJ distribution) and the [MorphoLibJ](https://imagej.net/MorphoLibJ#Installation) plugin.

# This script performs the following algorithm:

1. Either “mean” or “median” for *intracellular* averaging can be selected (“average”) using the options in the script
2. Select a cell , and find , which is the position that has maximum average intensity in channel 1.
3. It then implements the following formula at :

Where:  
 = ratio intensity of cell   
 = channel 1's average local background intensity   
= channel 2's average local background intensity  
 = channel 1's average local intensity inside a cell  
= channel 2's average local intensity inside a cell

1. Finally, produce a list of all for each of cells. This list is then used to calculate the mean, standard deviation and median intensity across all cells. Also outputs the number of cells (i.e. number of labels minus 1 for the background).

# Requirements

This script works with the Fiji build of ImageJ 1.51u with MorphoLibJ plugin installed. It requires:

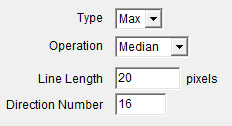
* A “data file” (e.g. a .lsm file) which contains two channels and can contain many z-slices.
* A “label file”, which is an image file (e.g. a .tif) which has cells “labelled” as a region of a single colour. The color is an integer between 0 and 255 called a “label”. Each cell must be labelled an integer between 1 and 255. The label of 0 is ignored entirely. The background is also assigned an integer label between 1 and 255, and is assumed by the label corresponding to the largest area by default.

# Script Usage

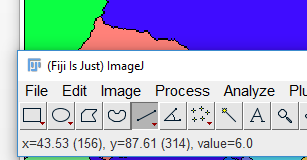
1. Simply open the script (**CalculateMeanIntensity.py**) in ImageJ/Fiji and open the “data file” and prepared “label file”. (The same label image will be used for all channels and z-levels.)
   1. If the largest label in the “label file” is not the background, then change the variable “**manually\_assign\_backgroundlayer\_to\_label=None**” to the background label
   2. By default “**warning\_threshold=0.01**” is set. This means that if is less than 0.01 then print a warning message. E.g. if the background is brighter than the cell on one of the channels a negative value will result, and a warning will be printed.
2. Click run. A Graphical User Interface (GUI) will pop up asking you to select which file is the “data file” and which is the “labels file”. Click OK and the program will run and print the results to the console.
   1. If you would like to run on many files and print the results to a .csv file, see the commented out code at the bottom of the script.
3. The script will output a summary.csv with the mean, standard deviation and median across all calculated average of all cells will be displayed, with each row being the file or time-step.

# Automatic Preparation of a label file

1. Open an image of the cells which is well-illuminated and shows the desired boundaries clearly. This can be a z/channel stack or a single image.
   1. If you want to select only one image from a z/channel stack simply Image>Duplicate and type in the desired channel and z level.
2. **Image > Type > 8-bit**
   1. (The reason for this is it makes analysis faster and means the default threshold value for the ‘MorphoLibJ Segmentation’ plugin appropriate
3. Apply either a Gaussian filter (**Process > Filters > Gaussian**) or a Median filter (**Process > Filters > Median**). A value of 2 pixels seems fine.
   1. I recommend a Gaussian filter
   2. (The reason for this is to remove noise in the output).
4. *Optional*: If a cell wall has gaps in it, we can fill them in using a **Plugin> MorphoLibJ > Morphological Closing** (<https://imagej.net/MorphoLibJ#Opening_and_closing>)
5. *Optional*: If the cell walls are really weak signal, then one option to make them brighter is to first perform **Plugin> MorphoLibJ > Directional Filtering**. This enhances the image while preserving the thickness (<https://imagej.net/MorphoLibJ#Directional_filters>) I found these settings worked quite well:



1. Use any of the four plugins (**Plugins> MorphoLibJ > Segmentation**) to obtain automatic labelling of cells.
   1. If the cells are very clear, you can use the (P**lugins> MorphoLibJ > Segmentation > Interactive-marker based Segmentation**), by clicking on each cell and then clicking run.
   2. But I found best results using (**Plugins> MorphoLibJ > Segmentation > Morphological Segmentation**)
      1. The reason for this is it has a tolerance parameter that can be varied – lower tolerance is equal to more labels, and higher tolerance is less labels.
      2. The other reason for this is that you can apply a gradient operation, which will look for edges and emphasise them. This may not be needed.
      3. If you have a z-stack file, you can also sweep z for the best labels.
   3. Finally, to save the result, click **Display > Catchment** Basins then click **Create Image**
2. We can visualise what number label that is assigned by hovering the mouse over a region, e.g. in the below example, the area the mouse is at has a label of 6.



1. It is likely some areas will be incorrectly labelled, and there will be some “tiny” labels that are not cells. To tidy up the image we can use tools in **Plugins> MorphoLibJ > Label Images** tools:
   1. **Plugins> MorphoLibJ > Label Images > Label Editions** provides a GUI for this.
      1. Merge labels into the same label**:** Click on two cells to mark them with a ROI and click **Merge** to combine their labels. Then hold ‘Alt-left click’ on each ROI to remove it.
      2. Remove labels which are small: Click on **size opening** and type the minimum area. Anything below this well be removed from the analysis (given a label of 0, indicated as black). Note: the background should have *not* be removed and can have any positive integer label.
      3. To remove cells on the edge: Click **remove in border**
   2. **Plugins> MorphoLibJ > Label Images > Replace/Remove Labels** allows you type change label numbers or remove labels. If you accidently delete the background so it becomes black (label=0), you should re-label it to a positive integer.

**How to manually label areas**

1. Ensure the image is 8-bit (**Image > Type > 8-bit**)
2. Select a tool to draw: 
3. Draw a shape
4. Press **Ctrl-D** (or **edit > draw**)
5. Choose a color (i.e. label) by pressing **Ctrl-Shift-K** (**Image > Color > Color picker**)
6. Use the flood-fill tool to fill the area 
   1. You may need to double-click on flood-fill tool to ensure it is set to “4-connected”
   2. It should now be labelled with an integer between 1 and 255.