Image Processing Tool for Quantification of Cells in Fibrotic Tissue

Files you will find here:

- BullseyeProcessing.py
- ExcelMaker.py
- GUI.py
- ilastikFull.m
- image_slicer.m
- Lipowski_FinalCapstoneReport_Group9.pdf
- UserManual.pdf

First, read **Lipowski_FinalCapstoneReport_Group9.pdf** to familiarize yourself with the context of this tool and why it was made.

To make image processing more manageable (if your images are large), divide them up int smaller “tiles” to make the image processing less computationally expensive. Use **image_slicer.m** to slice the image up into smaller files. This step is optional if your image sizes are not large or you have a lot of computing power available. Note: in the file you must change the directory to the corresponding directory where the image sits and where you want your “tiles” to sit.

Load Ilastik (https://www.ilastik.org/).

Select the ‘Pixel Classification’ option in the main menu.

There are 5 steps Ilastik requires:

1. Input data. Select a few sample images. This is your training data
2. Select features. The more you select the better performance you can expect.
3. Training. Add a label and draw on the image to the right to label objects of interest. Use the ‘current view’ to cycle through different samples used for training. Click ‘simple segmentation’ to preview the segmentation of objects in the image until you are satisfied.
4. Prediction export. In the case of immunostained images, ensure that the export setting are set to the following. Output directory will vary.

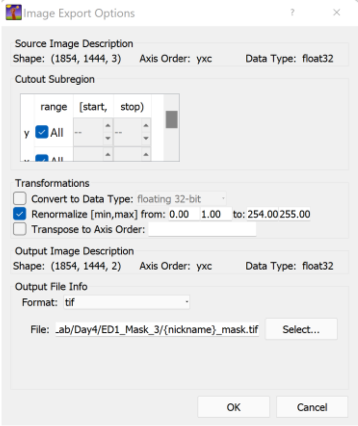

Image Export Options

Source Image Description
Shape: (1854, 1444, 3) Axis Order: yxc Data Type: float32

Cutout Subregion

	range	[start,	stop)
y	☑ All	--	--
x	☑ All		

Transformations
□ Convert to Data Type: floating 32-bit
☑ Renormalize [min,max] from: 0.00 1.00 to: 254.00 255.00
□ Transpose to Axis Order:

Output Image Description
Shape: (1854, 1444, 2) Axis Order: yxc Data Type: float32

Output File Info
Format: tif
File: _ab/Day4/ED1_Mask_3/{nickname}_mask.tif

5. Batch processing. Select raw data files (images) from a directory (the one that tiles were saved to in image_slicer.m) and process all files. The outputs will be found in the output directory specified earlier.

Use **ilastikFull.m** to stitch all output from Ilastik together. Again, ensure corresponding file names for the Ilastik outputs are adjusted. This is necessary for annular analysis to be competed.

The front-end user interface utilized three files: **GUI.py, ExcelMaker.py, and BullseyeProcessing.py**.

To run the program, simply run **GUI.py**. Ensure that all file directories are properly identified under the “TODO” at the top of the GUI.py script. Note: line 15 allows for the inclusion of a logo in your GUI. Ensure you have a file named ‘Logo.png’ with your organization’s logo.

ExcelMaker.py is used to organized all data into an excel spreadsheet. Please edit how you see fit.

BullseyeProcessing.py is responsible for annular analysis. Please edit any filename nomenclature conventions as you see fit.