Watershed Cells GUI

by Lena R. Bartell

Table of contents

- Table of contents
- Installation
  - Download the GUI files
  - Add the GUI folder to your MATLAB path
  - Check your MATLAB version
- Usage
  - Start the GUI
  - Select Images
  - Segmentation
  - Classification
  - Save Data
  - Batch Process
- More Information
  - Algorithm Details
  - Feedback

Installation

Download the GUI files

Download the GUI (graphical user interface) files to a folder and save it somewhere. The folder should contain the following files:

```
watershed_cells_gui.m
watershed_cells_gui.fig
private\apply_function.m
private\apply_threshold.m
private\find_regions.m
```

You may also want to include the provided example image, `test.tiff`.

Add the GUI folder to your MATLAB path

In MATLAB, go to the HOME tab and, in the ENVIRONMENT section, click “Set Path”. In the window that opens, click “Add Folder…” and add the folder containing the GUI files to your path.

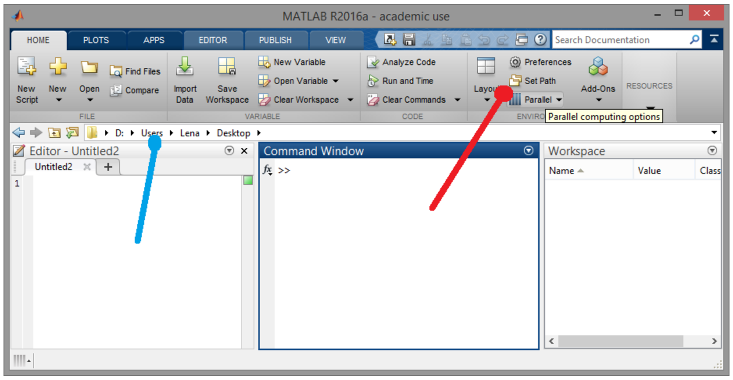


Figure 1. MATLAB window highlighting Set Path (red arrow) and current folder (blue arrow).

Check your MATLAB version

This GUI will only work on MATLAB R2014b or later and requires the Image Processing Toolbox. To check your version of MATLAB, enter `ver` into the Command Window. You should get something that looks like this:

```
>> ver
-----------------------------------------------------------------------
MATLAB Version: 9.0.0.341360 (R2016a)
...
-----------------------------------------------------------------------
MATLAB                                    Version 9.0         (R2016a)
Image Processing Toolbox                  Version 9.4         (R2016a)
...
```

Note: This GUI has only been tested on Windows machines. I have no expectation that it will work on other operating systems.

Usage

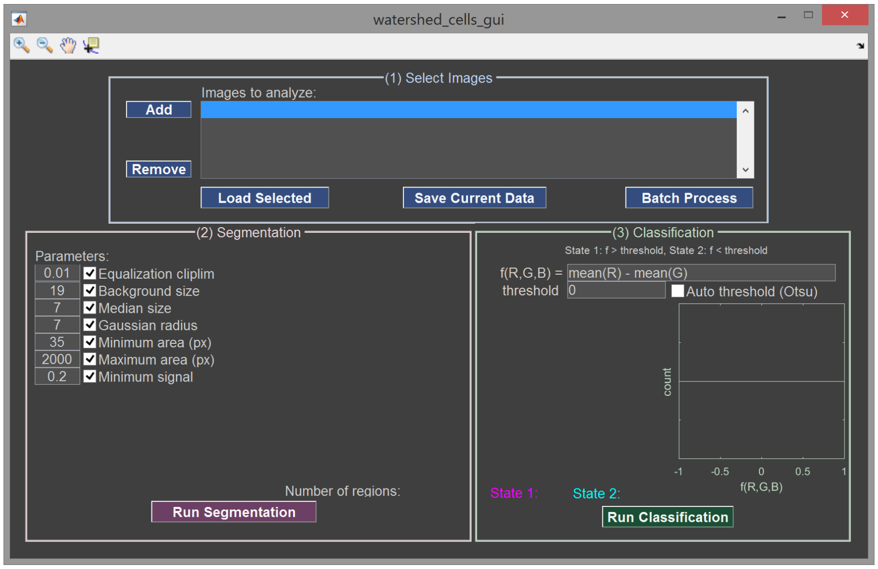
Start the GUI

To start the watershed cells GUI, run:

```
>> h = watershed_cells_gui
```

This will open the watershed cells GUI in a window with handle `h`. Within this handle, you can access the current state of the GUI data in the `UserData` field. In particular, `h.UserData` is a struct with the field `params`, which holds the current paramers from the GUI and `results`, which holds the current analysis results. For example, after the GUI opens, run `h.UserData.params.image.path` to return the path to the image being analyzed, or run `h.UserData.results.segmentation.number` to return the total number of regions found during segmentation. These fields are all populated with default values when the GUI is created, but will update as you use the GUI.

When the GUI first opens, it should look like this:



Images to analyze:

Parameters:
0.01 ☑ Equalization cliplim
19 ☑ Background size
7 ☑ Median size
7 ☑ Gaussian radius
35 ☑ Minimum area (px)
2000 ☑ Maximum area (px)
0.2 ☑ Minimum signal

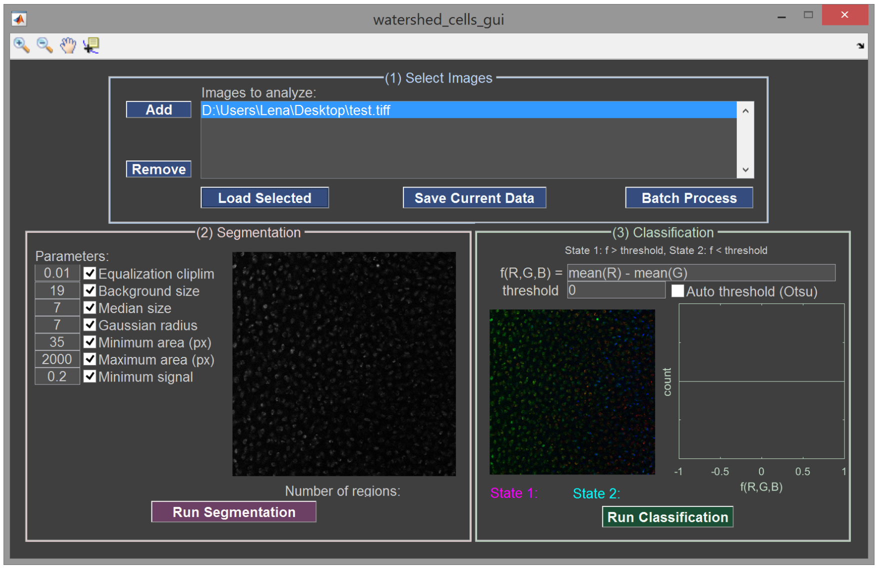
Number of regions:

f(R,G,B) = mean(R) - mean(G)
threshold 0 □ Auto threshold (Otsu)

Figure 2. watershed_cell_gui initial appearance

Select Images

In the `Select Images` panel, click the `Add` button to open a dialog box and select an image you want to process. To follow this example, use the included `test.tiff` file. Click `Load Selected` to import and display the selected image. The `Segmentation` panel shows a grayscale version of the image used for finding regions, whle the `Classification` panel shows a full color version used for segmentation. At this point, the GUI should look like Figure 3. You can use the toolbar buttons to zoom/pan and inspect the images.



Images to analyze:
D:\Users\Lena\Desktop\test.tiff

Parameters:
0.01 ☑ Equalization cliplim
19 ☑ Background size
7 ☑ Median size
7 ☑ Gaussian radius
35 ☑ Minimum area (px)
2000 ☑ Maximum area (px)
0.2 ☑ Minimum signal

Number of regions:

f(R,G,B) = mean(R) - mean(G)
threshold 0 □ Auto threshold (Otsu)

Figure 2. watershed_cell_gui after importing an image

Segmentation

In the GUI, set the segmentation parameters by editing the text boxes in the `Segmentation` panel. To skip a particular step in the analysis, un-check the tick box next to the associated parameter. In general, you will need trial-and-error to pick the best parameters for your image. However, there are some rough guidelines below.

The parameters (and guidelines) are:

- *Equalization cliplim*: This is the “Clip Limit” parameter passed to the adaptive histogram equalization filter. This is a number between 0 and 1, but values around 0.01 are usually best. See MATLAB’s function `adapthiseq` for more information.
- *Background size*: Background subtraction is performed using a median filter of this size (units: pixels). This should be an odd positive integer that is much larger that the diameter of your cells/objects. Background subtraction is usually the slowest step in the process,

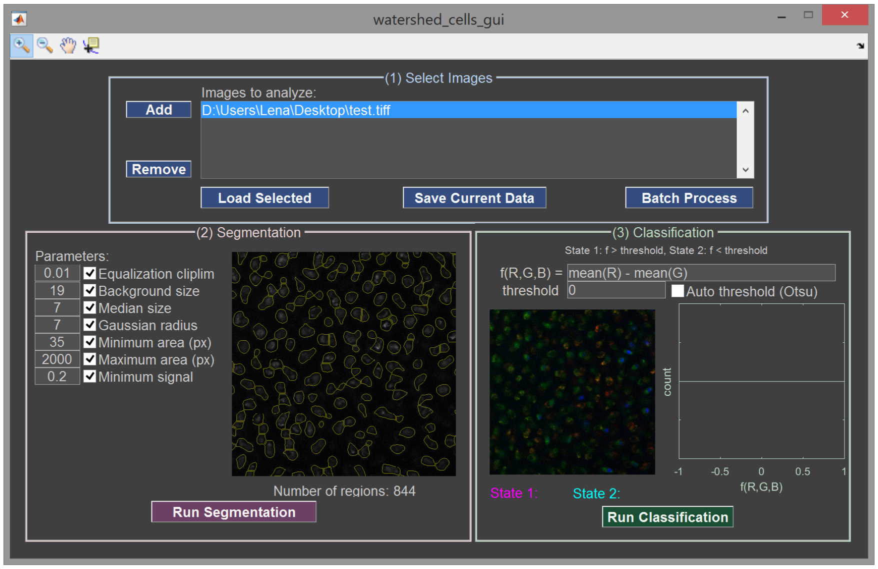
so skip it if you don’t need it. Generally, background subtraction important for epi-fluorescence images (due to unwanted out-of-focus/background signal), but optional for confocal images (because the confocal pinhole already blocks much of this background).

- *Median size*: The size of a separate median filter used for smoothing / reducing noise (units: pixels). This should be roughly the same size or smaller than the cells you are trying to find.
- *Gaussian radius*: The radius of a Gaussian filter, which is also used for smoothing / reducing noise (units: pixels). If the signal is discontinuous across an individual cell (e.g. if staining individual organelles rather than the cytoplasm), this radius should be approximately the radius of a cell. If the signal is relatively continuous across the cell (e.g. standard live/dead staining assay), this can be smaller. Values smaller than 0.5 px are not recommended.
- *Minimum area*: After finding objects/cells, discard any objects that have fewer than this many pixels in their area (units: pixels). This should be approximately r2, where r is the average cell radius in pixels.
- *Maximum area*: Also discard any objects that have more than this many pixels in their area. This should be approximately (2r)2 to (5r)2, where r is the average cell radius in pixels.
- *Minimum signal*: Also discard any objects that have an average signal-intensity smaller than this value (units: fraction of intensity range). This value is expressed as a fraction of the full range of possible intensities, so it should be a number between 0 and 1 (rather than 0 and 255, for example). This value will vary drastically based on imaging settings, image quality, other parameter values, and your desired output, so I won’t provide any guidelines.

Once parameters are set, click `Run Segmentation` to run the image analysis process and display the result. Briefly, the image analysis procedure is:

1. Pre-processing
   i. Import image
   ii. Adaptive histogram equalization
   iii. Background subtraction using a median filter (using "background size" parameter)
   iv. Median filter to smooth
   v. Gaussian filter to remove noise
2. Watershed segmentation
   i. Determine background using a conservative Otsu's threshold
   ii. Watershed segmentation, with background enforced
3. Post-processing
   i. Remove cells that are too small
   ii. Remove cells that are too big
   iii. Remove cells that are too dim

Once the image segmentation is complete, regions that are found will be circled in yellow on the plot, as in Figure 4. You can use the toolbar buttons to zoom/pan and inspect the plots.



watershed_cells_gui

(1) Select Images

Images to analyze:
D:\Users\Lena\Desktop\test.tiff

(2) Segmentation

Parameters:
0.01 ☑ Equalization cliplim
19 ☑ Background size
7 ☑ Median size
7 ☑ Gaussian radius
35 ☑ Minimum area (px)
2000 ☑ Maximum area (px)
0.2 ☑ Minimum signal

Number of regions: 844

(3) Classification

State 1: f > threshold, State 2: f < threshold

f(R,G,B) = mean(R) - mean(G)
threshold 0 □ Auto threshold (Otsu)

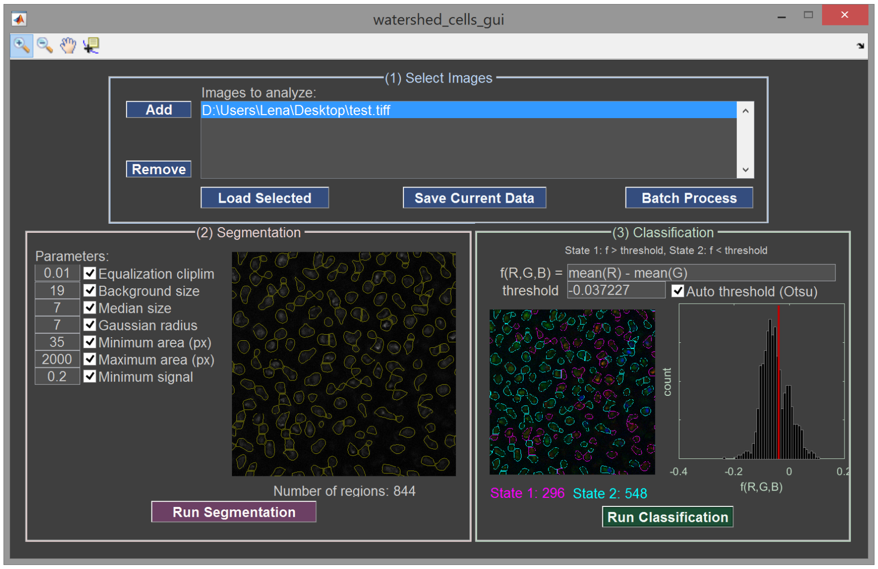
Figure 4. watershed_cell_gui segmentation result. Note that the number of regions found is also shown on the screen.

Classification

To classify regions, the GUI applies a function, `f(R,G,B)` to each region, where R, G, and B, are, respectively, the list of red, green, and blue pixel values in the given region. This function should return one scalar number for each region. This number will then be thresholded according to the `threshold` parameter. If `Auto threshold (Otsu)` is selected, then the threshold value will be chosen automatically based on Otsu's thresholding scheme.

Enter the function definition and threshold (if manual) in the text boxes. The function may be a custom function and/or may call other Matlab and user-defined functions. Then, click `Run Classification` to run the classification process and display the results.

Onces the classification is complete, the regions with f values above and below the threshold will be outlined in magenta and cyan, respectively. Also, the GUI shows a histogram of the f values from all the segmented regions. You can use the toolbar buttons to zoom/pan and inspect the plots. At this point, the GUI should look like Figure 5. You can use the toolbar buttons to zoom/pan and inspect the plots.



Images to analyze:
D:\Users\Lena\Desktop\test.tiff

Parameters:
0.01 ☑ Equalization cliplim
19 ☑ Background size
7 ☑ Median size
7 ☑ Gaussian radius
35 ☑ Minimum area (px)
2000 ☑ Maximum area (px)
0.2 ☑ Minimum signal

Number of regions: 844

State 1: f > threshold, State 2: f < threshold
f(R,G,B) = mean(R) - mean(G)
threshold -0.037227 ☑ Auto threshold (Otsu)

State 1: 296 State 2: 548

Figure 5. watershed_cell_gui classification result. Note that the number of regions above/below the threshold (state 1 / state 2) is also shown on the screen.

Save Data

Click `Save Current Data` to save the results currently shown in the GUI. This will open a dialog box asking the user to choose a folder in which to save the results. In this folder, the GUI will save two files (note that both files use the base name from the image file): 1. `<image name>_data.mat` : A MATLAB data file containing the structure 'data'. This is the same structure output at the command line (i.e. `data` from `[h, data] = ...`). This structure contains the field `params` , which specifies the parameters from the GUI's current state, and `results` , which contains the results of the segmentation and classification. 2. `<image name>_display.tif` : A TIF image file showing the imported image with all regions outlined based on their state (above threshold = magenta, below threshold = cyan, equal to threshold = yellow). Open this image to visually check the results.

Batch Process

You can also use the GUI to automatically process multiple images with the same parameters. To do this first use the `Add` button to select all the files you wish to process. Next, setup all the parameters in both the `Segmentation` and `Classification` panels. Finally, click `Batch Process` to analyze all the listed images. This will loop through each listed image file, load it, segment it, classify it, and save resulting data, before moving on to the next image on the list.

As the bath process continues, the GUI plots will dynamically update to show the current processing results. While the batch process is running click the `Cancel` button to stop the analysis. This will finish completing the current calculation step and then cancel all remaining steps. Any unsaved data will be lost.

More Information

Algorithm Details

The “meat” of the segmetnation algorithm is encoded in the private function `private\find_regions.m`. If you would like to alter the processing algorithm or use it externally or programmatically, start there. Similarly, the "meat" of the classification process is contained in `private\apply_function.m` and `private\apply_threshold.m`.

Feedback

I welcome feedback. If you run across any errors or issues, please email me the details and I will try to help. Similarly, please let me know if you find this GUI useful. Good news is always appreciated! :)