How to run it and what to expect

Thursday, January 13, 2022 3:32 PM

How to run the code (going line by line down the code for things to change):

- Change the function whose return value is stored in 'ip'. This should be at the start and called "exp_YYYYMMDD_InitializeExp;"
- Tell the code which fluorescent channels are which channel index.
- Make sure 'norm_over_all' is set to 0. This way each cell is normalized over itself rather than with respect over the whole fluorescent image.
- Set the 'combo' of 'high_threshold' and 'low_threshold' to be run. The reason why a histogram is produced for each image is so that you can go back and decide for yourself a good value of high and low threshold to place here. This also makes it easy to test a few thresholds all at once.
- Make sure the 'spot_folder' is how you wish for it to be named/organized.
- Check the function file "InitializeClusterRecognitionParameters_c234" to make sure it points to the correct file locations for the microscope and segmentation images.
- Make sure the disk radii for the 'imerode' and 'imdilate' functions are what you want them to be.

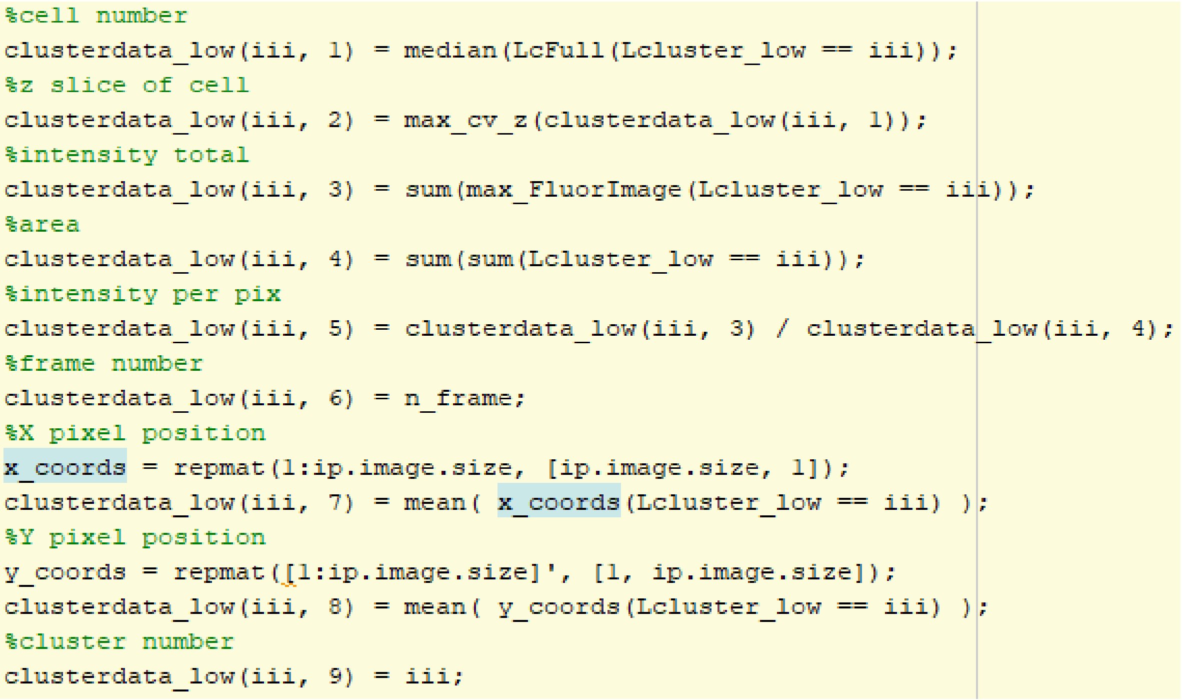
What to expect as outputs:

As mentioned previously, for each image we will be returned a histogram of pixel intensities with a three gaussian curve fit. This three gaussian curve fit is supposed to approximate a three pixel intensity population within the cell: Background, low threshold area, and high threshold area.

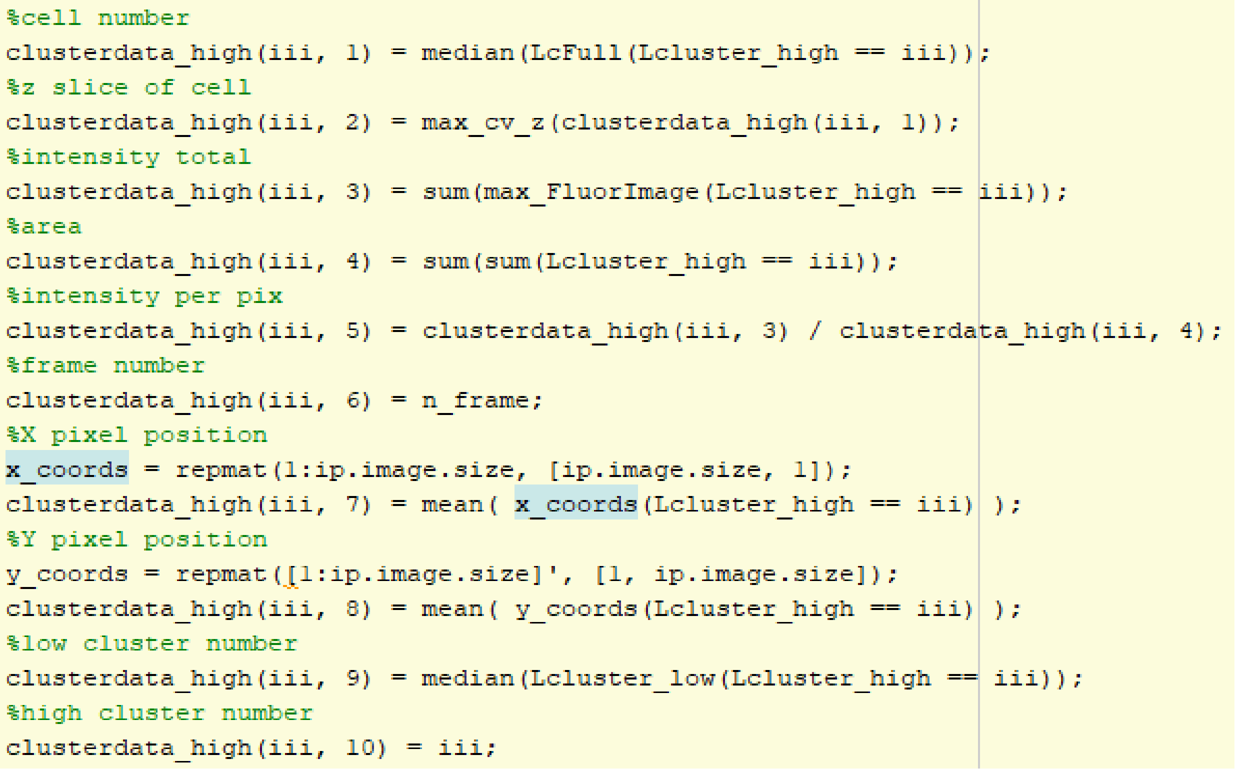
For each image we will also be returned a reconstruction of the normalized fluorescent image, the threshold objects without filters, and the final filtered objects. More detail into the displaying algorithm can be found in the 'Displaying Images' tab.

Finally, we are also returned data regarding the threshold objects:

- A matrix of enumerated low threshold objects is stored for each image (I have elected to REMOVE the high threshold areas, this can be changed by commenting out the following line:
    - " Lcluster_low = Lcluster_low .* (Lcluster_low & ~smooth_T_high_image); "
- A matrix of enumerated high threshold objects is stored for each image.
- Below is what is returned for the low threshold objects and how they are extracted from the images:

- 

```
%cell number
clusterdata_low(iii, 1) = median(LcFull(Lcluster_low == iii));
%z slice of cell
clusterdata_low(iii, 2) = max_cv_z(clusterdata_low(iii, 1));
%intensity total
clusterdata_low(iii, 3) = sum(max_FluorImage(Lcluster_low == iii));
%area
clusterdata_low(iii, 4) = sum(sum(Lcluster_low == iii));
%intensity per pix
clusterdata_low(iii, 5) = clusterdata_low(iii, 3) / clusterdata_low(iii, 4);
%frame number
clusterdata_low(iii, 6) = n_frame;
%X pixel position
x_coords = repmat(1:ip.image.size, [ip.image.size, 1]);
clusterdata_low(iii, 7) = mean( x_coords(Lcluster_low == iii) );
%Y pixel position
y_coords = repmat([1:ip.image.size]', [1, ip.image.size]);
clusterdata_low(iii, 8) = mean( y_coords(Lcluster_low == iii) );
%cluster number
clusterdata_low(iii, 9) = iii;
```

- Below is what is returned for the high threshold objects and how they are extracted from the images:
  - 

```
%cell number
clusterdata_high(iii, 1) = median(LcFull(Lcluster_high == iii));
%z slice of cell
clusterdata_high(iii, 2) = max_cv_z(clusterdata_high(iii, 1));
%intensity total
clusterdata_high(iii, 3) = sum(max_FluorImage(Lcluster_high == iii));
%area
clusterdata_high(iii, 4) = sum(sum(Lcluster_high == iii));
%intensity per pix
clusterdata_high(iii, 5) = clusterdata_high(iii, 3) / clusterdata_high(iii, 4);
%frame number
clusterdata_high(iii, 6) = n_frame;
%X pixel position
x_coords = repmat(1:ip.image.size, [ip.image.size, 1]);
clusterdata_high(iii, 7) = mean( x_coords(Lcluster_high == iii) );
%Y pixel position
y_coords = repmat([1:ip.image.size]', [1, ip.image.size]);
clusterdata_high(iii, 8) = mean( y_coords(Lcluster_high == iii) );
%low cluster number
clusterdata_high(iii, 9) = median(Lcluster_low(Lcluster_high == iii));
%high cluster number
clusterdata_high(iii, 10) = iii;
```

  - Note that the only difference is the addition of the final column labeling the high cluster.