PCD Quantification User’s Manual

MATLAB version: R2014a or higher

**3D-SIM quantification**

**Methods:**

1.1 Download all the MATLAB scripts to the local folder and set path for the folder.

1.2 Open the script ‘PCD\_Diagosis\_Main.m’ in MATLAB.

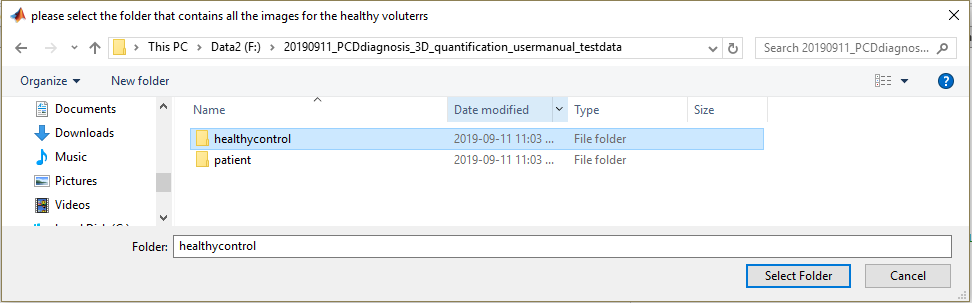
1.3 Modify the parameters at line 45 and line 46 to fit your data. See engineering problems for how to set the thresholds.

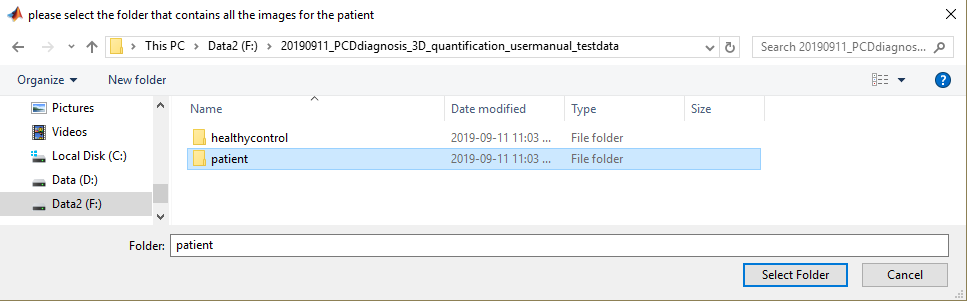
*% 3. Set the threshold for the red channel(cilia marker) and green channel(protein of interest)*

*threshold\_red=1000;*

*threshold\_green=500;*

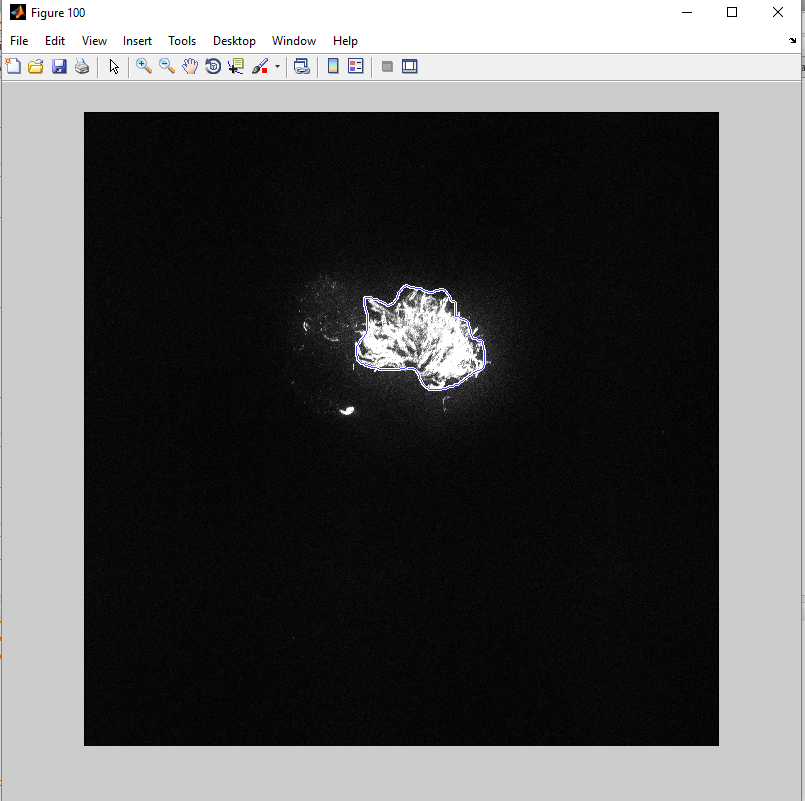
1.4 Run the script, select the folder that contains all the data for the healthy controls and the data for the PCD patients.



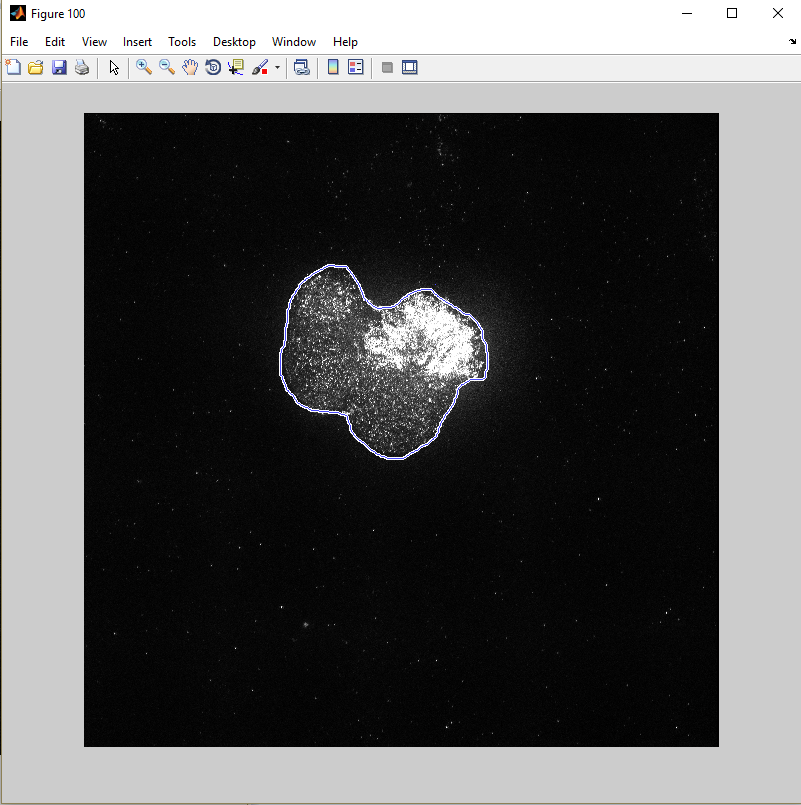


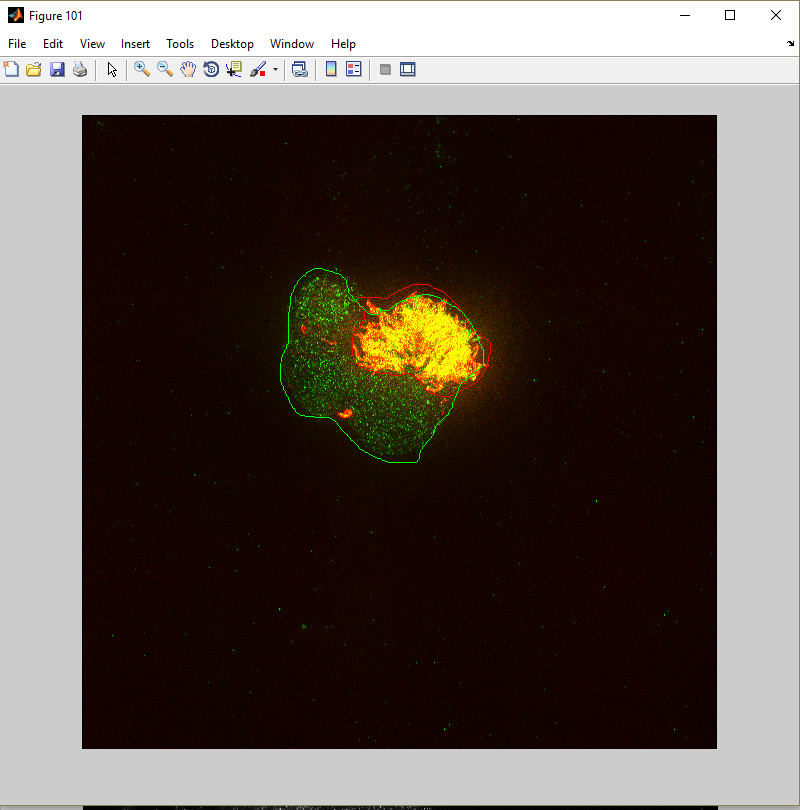
1.5 Wait the pop out and start to crop cell.

1.5.1 Define the boundary of the cilia first

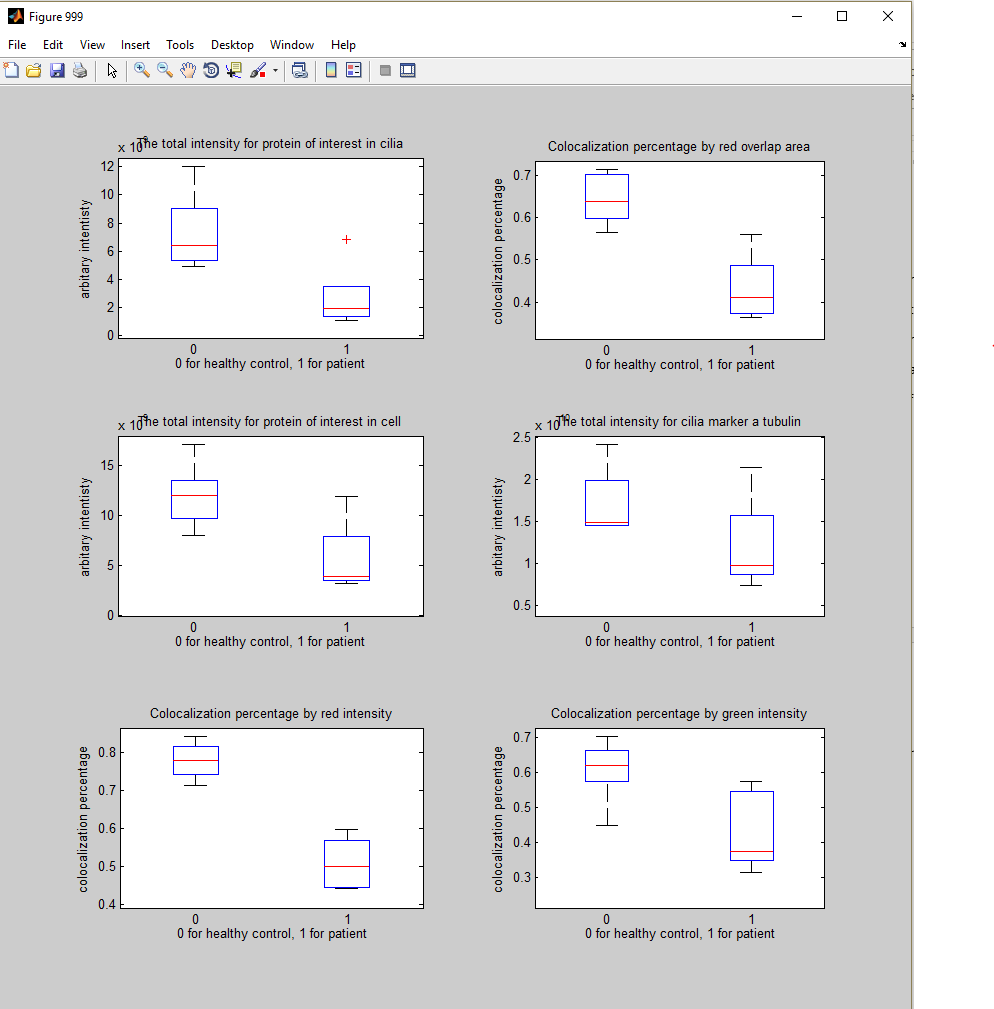


1.5.2 Define the boundary of the PCD signal secondly





1.6 repeat cropping cells for all the healthy control samples and patients. After you finish the cropping, the result will be shown as below and the corresponding data will also be save automatically.



Engineering problems:

1. How is the threshold determined?

The threshold is determined based on the mean intensity projection of the image. To generate the mean intensity projection image, you need to run the script “SIM\_3D\_mean\_intensity\_projection.m” under the folder. This script allows you select a folder and generates mean intensity projection for all the images under the folder. I will open a few wild type images and check either the atubulin or the PCD protein intensity in the cilium and use the intensity as thresholds.

2. What if there are multiple cells that I want to crop?

Currently, the script doesn’t support cropping two cells in one field of view, but you can always duplicate the image and crop the cell in the second copy of the image.

3. Is there an automatic way that saves me the trouble of cropping cells manually?

Unfortunately, there are no way now, it is still under construction.

**Machine learning for PCD diagnosis**

For easy implementation, it is suggested to visit the machine learning at google colab: "<https://colab.research.google.com/drive/17nRJfqF_AH1dLQL5fFkS8YYvCRuYtzOJ#scrollTo=EtaR6MGq8LX2&forceEdit=true&sandboxMode=true>"

By clicking the run button for each part, the script will lead you to the final result.



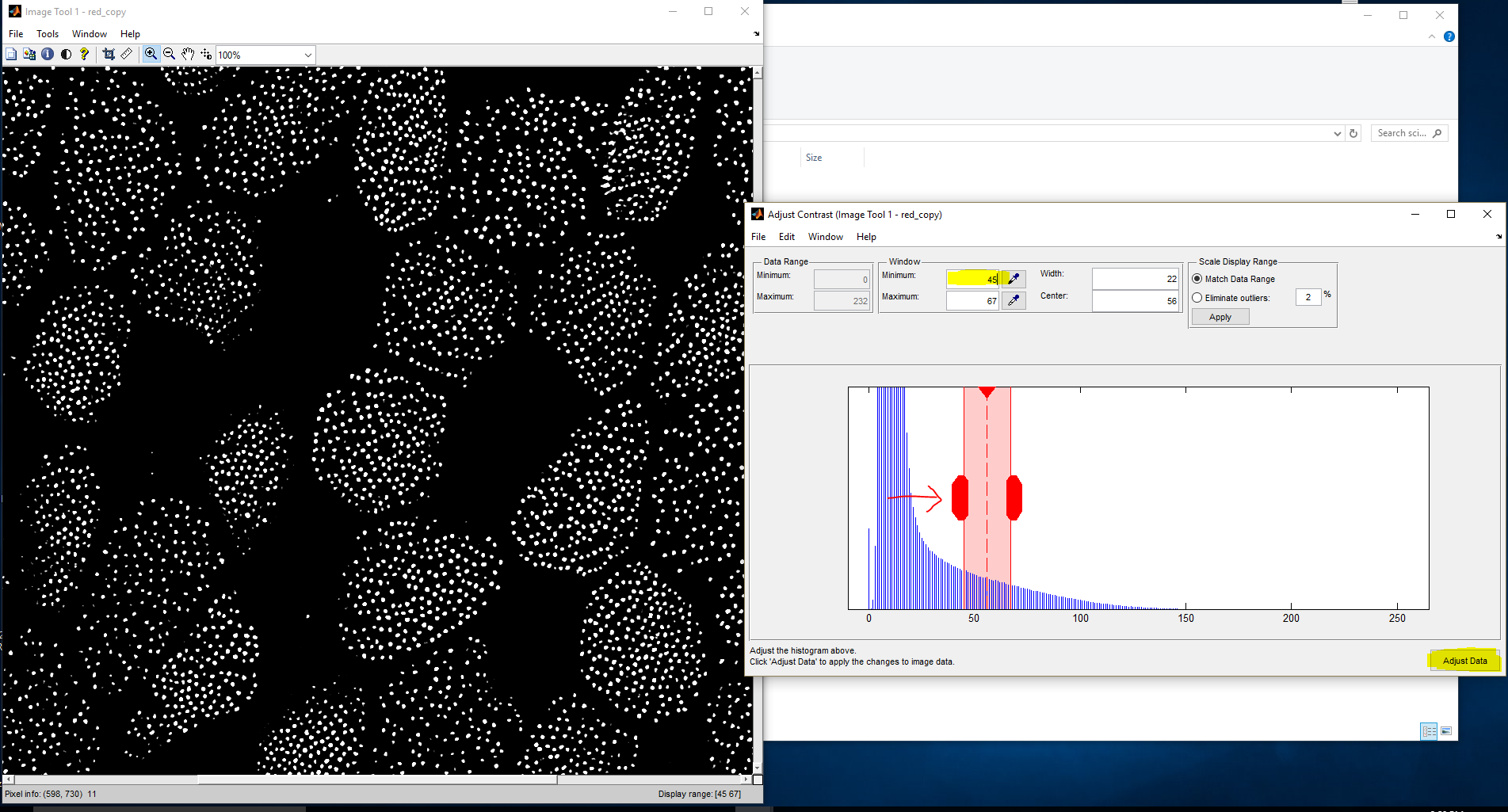
Notes: you need to download test data to your google drive root folder as the script need to visit this data. The test data for machine learning could be found at "https://drive.google.com/open?id=1ZnFG6HqncuswQevkj3wSeX0WmsvD44Jr"

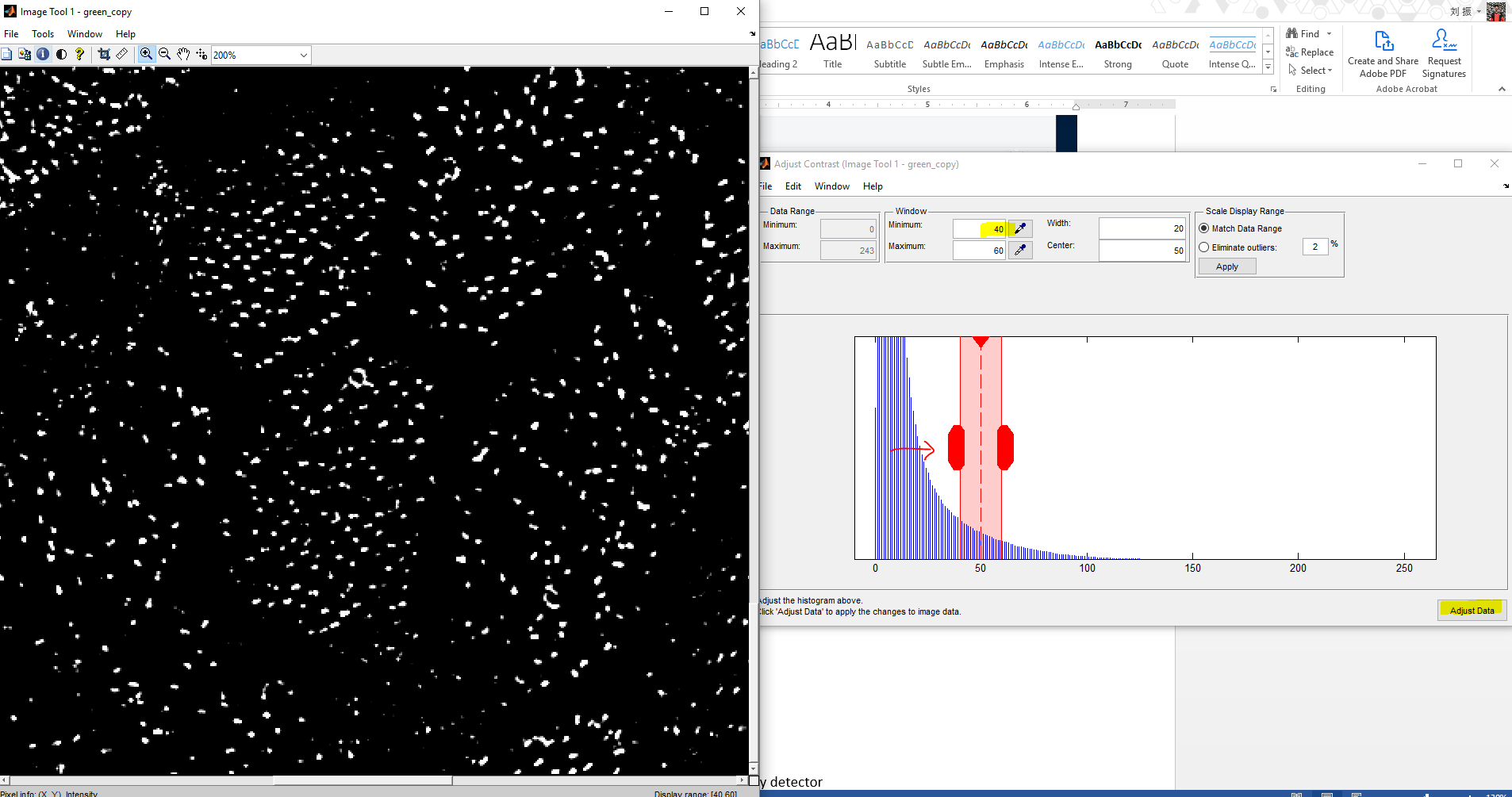
**Rotational Polarity Analysis**

**Methods:**

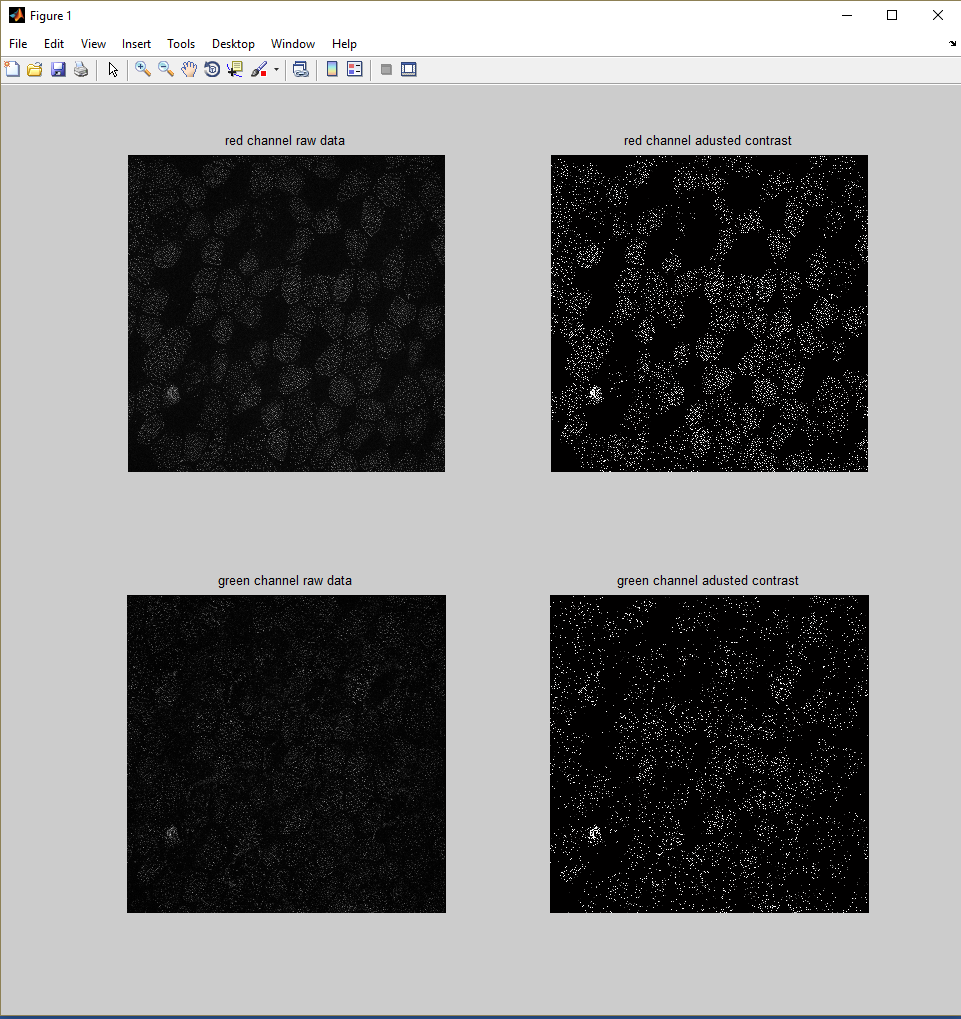
1. In MATLAB, run ‘RotationalPolarityAnalysis\_Main.m’, the script will guide you to select the image you want to analyze.

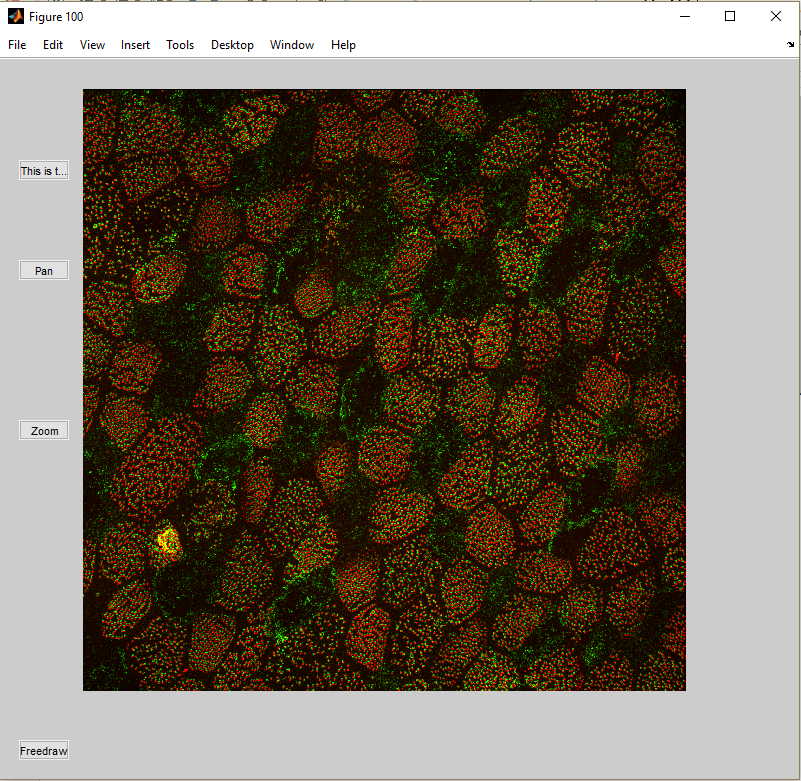
2. Set the thresholds for the red and green channel. Click the ‘Adjust data’ and close the right figures.





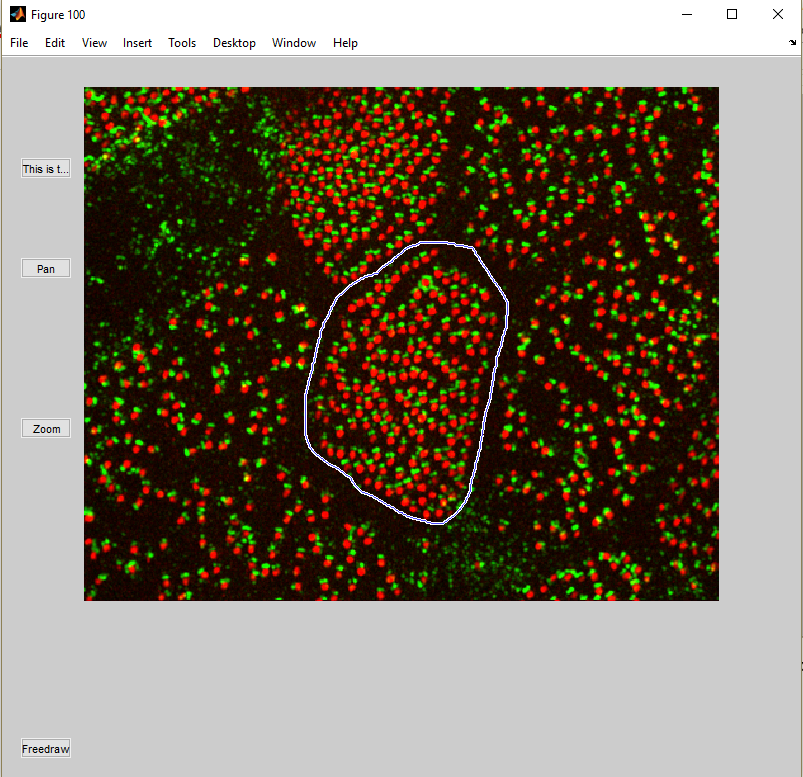
3. The MATLAB script will then show the binary counterpart for the whole image and guide you to the main interactive interface as show below.



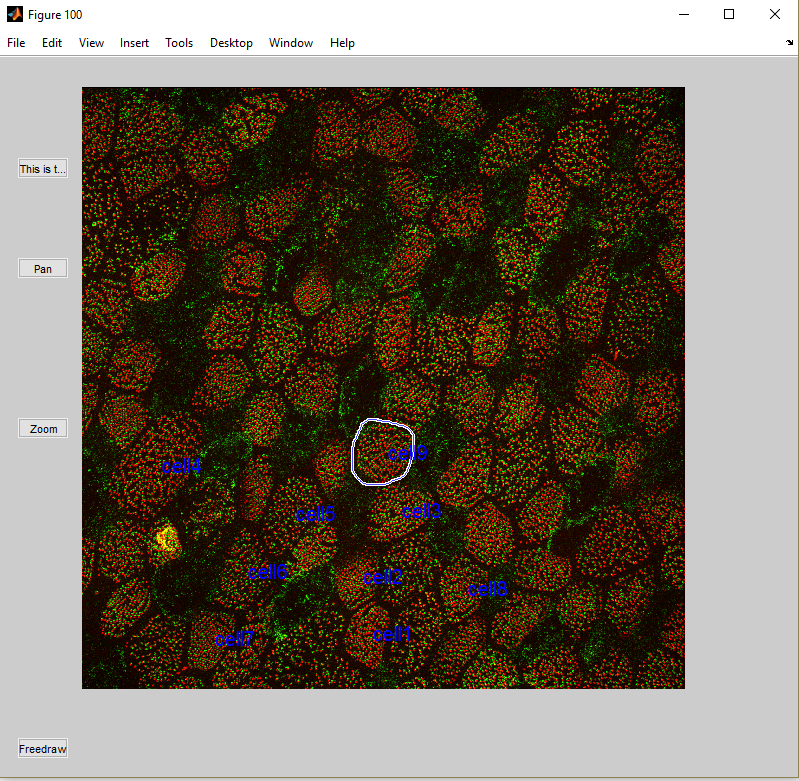


As you can see, there are a lot of useful buttons on the left. ‘Pan’ allows you to move a cell of particular interest in the center, ‘Zoom’ allows you to zoom in to check the details of one cell. ‘Freedraw’ is the one allowing you to manually define one cell. When you finish draw, several figures will pop out which are ‘binary images of the cell’, ‘identified direction pairs’, ’histogram of all directions identified’ and ‘basal body numbers’.

If you only have one cell of interest left, click ’This is the last cell’ and script will finish and exit. By doing this, the script will label all the cells you have cropped to keep track of all the information.







4. To get the vector length of the whole field of view or different cells from different repeats, put all the generated data into one folder, and run the script ’RotationalPolarityAnalysis\_Step2\_CalculateVectorLength.m’ and the command line will show the mean and median vector length.

*------------------------------*

*mean vector length is:*

*0.72695*

*------------------------------*

*median vector length is:*

*0.74647*

*------------------------------*

*vector length std is:*

*0.11908*

*------------------------------*

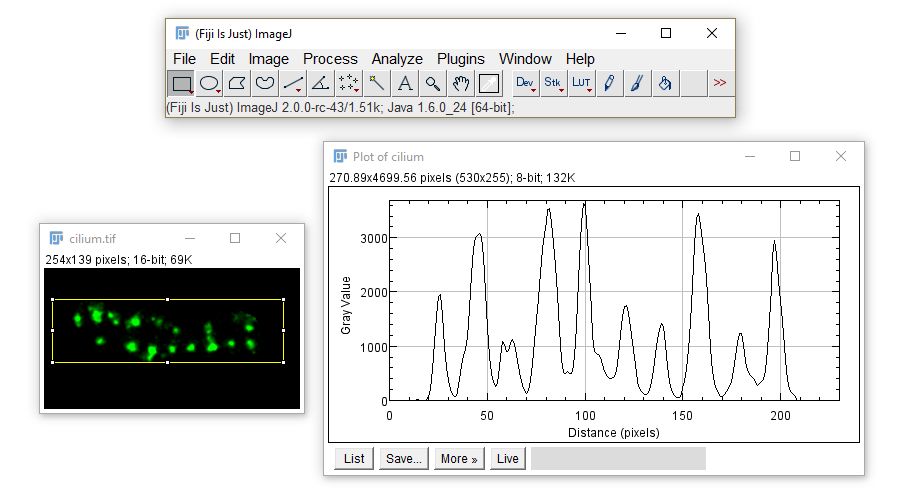
**Engineering problems:**

1**. What is the cutoff for the healthy control and PCD patients?**

2. What if I forget to click “This is the last cell” and use the ‘freedraw’ to crop the last cell?

It is fine. You just need to click ‘This is the last cell’ and crop the cell again. Then the cells will be counted twice in the whole field view vector length calculation.

4. Periodicity detector

**Methods**

1. Open the “cilium.tif” file by image.

2. Define a rectangle by the “rectangle tool” in the toolbar.

3. Click ‘Analyze’->’Plot Profile’, you will get the intensity distribution along the cilium.

4. Click ‘Save’ in the figure and save the intensity distribution to a txt file.

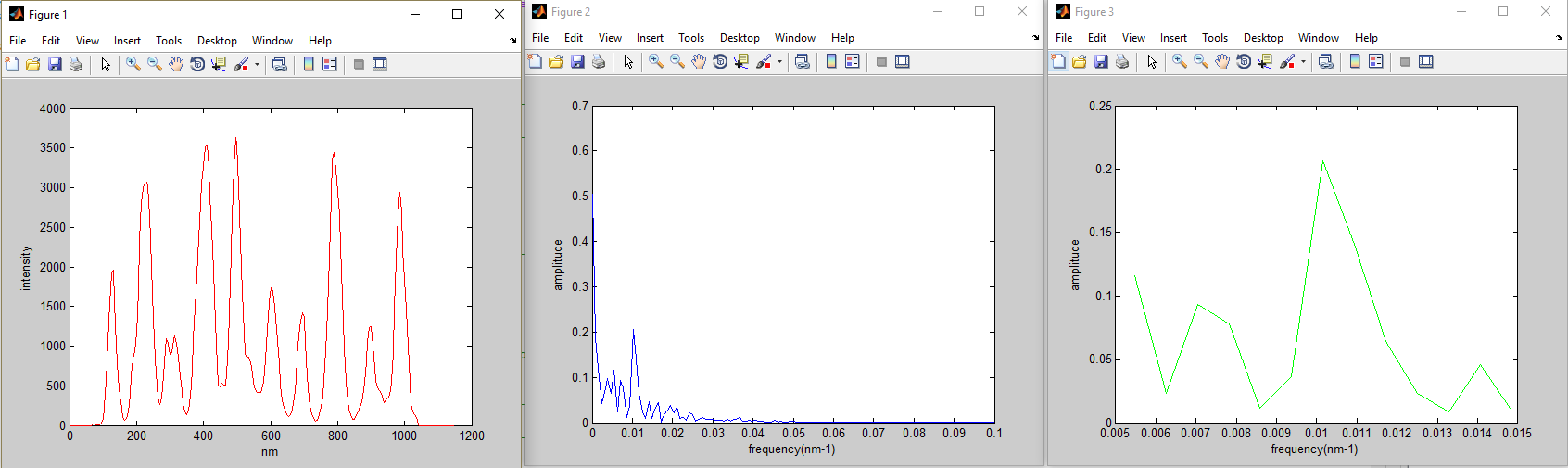
5. In MATLAB, run the ‘’fourier\_transform\_main.m” script. Choose the txt file you just saved and three figures will pop up when it is finished. The commander window will show the periodicity as below:

*----------------------------------------------------------------*

*The periodicity is:*

*98.4615nm*

*----------------------------------------------------------------*



**Engineering problems:**

1. What if the intensity distribution is not periodic?

The selection of the right cilium is critical. It is recommended that the selected cilium is straight rather than bent as bent cilium might twist the intensity distribution. One more thing is that the cilium show points horizontally rather than vertically.

2. What if the scale is not correct?

The default scale here is 5 nm per pixel. If your images doesn’t obey the scale, go to line 9 of the script and change 5 to your pixel size.

*----------------------------------------------------------------------*

*data(:,3)=data(:,1)\*5; % 1 pixel=5 nm*

*----------------------------------------------------------------------*