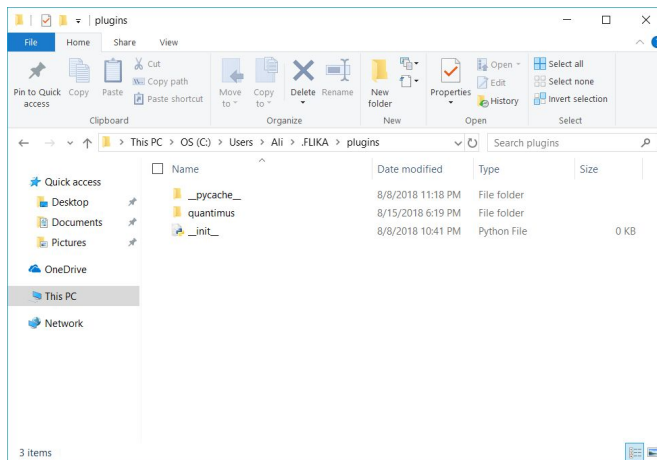


Installation:

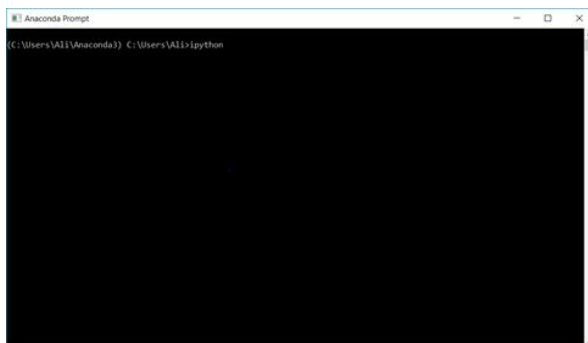
1. Follow initial installation process at <https://quantimus.github.io/documentation.html>
2. Download QuantiMus repository from <https://github.com/Quantimus/quantimus>, unzip it, and rename to 'quantimus'.
3. After doing so, move the folder to the Flika folder shown in the image below. Folders that begin with '.' are often hidden. Please make sure your computer settings allow these files to be viewed (this differs for different operating systems).



Note: we are currently working on an easier way to download QuantiMus, please bear with us as we add an option to download QuantiMus directly from Flika!

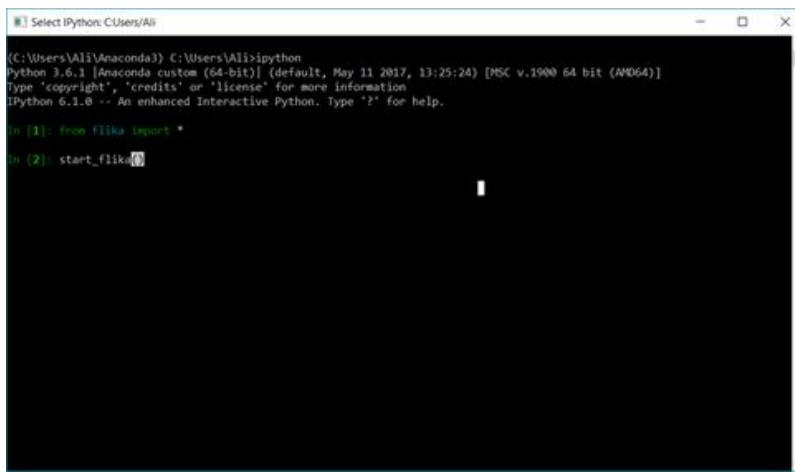
Starting up:

1. Open “Anaconda Prompt,” and type: “**ipython**”. Click enter/return to enter the ipython shell



2. Once in the ipython shell, type: “ **from flika import *** ”. This will import all required Flika packages and software. Click enter, and then type “ **start_flika()** ”; click enter to open Flika.

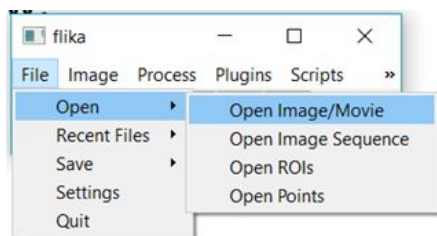
Though no more functions will be run using this terminal, the user should keep this window open at all times Flika is being used. The terminal will also be used to display progress of QuantiMus, as well as properties of regions of interests that are clicked in later stages of the algorithm.



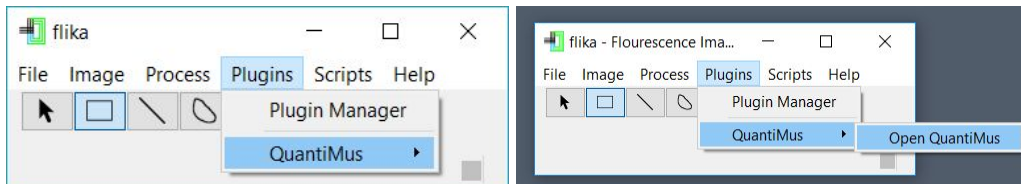
```
Select IPython: C:\Users\Ali
(C:\Users\Ali\Anaconda) C:\Users\Ali>ipython
Python 3.6.1 [Anaconda custom (64-bit)] (default, May 11 2017, 13:25:26) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from flika import *
In [2]: start_flika()
```

3. To select an image, click: *File -> Open -> Open Image/Movie*



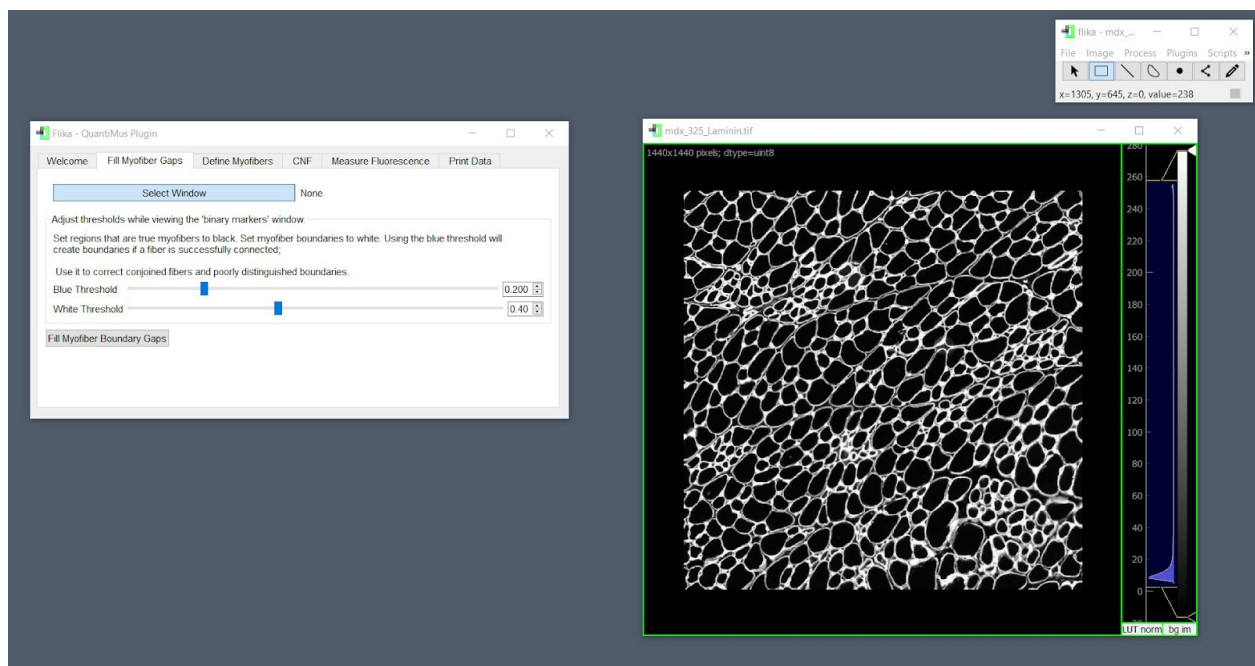
4. Once an image is opened, open QuantiMus plugin: *Plugins -> Quantimus -> Open QuantiMus*

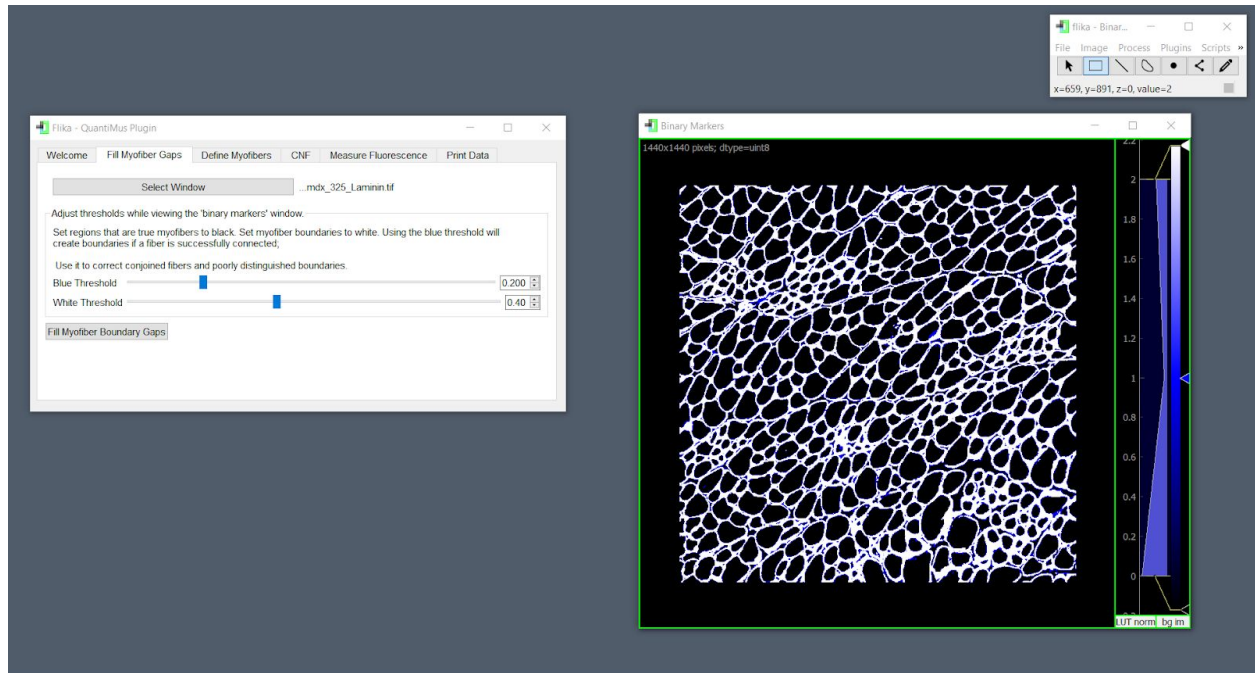


QuantiMus analysis:

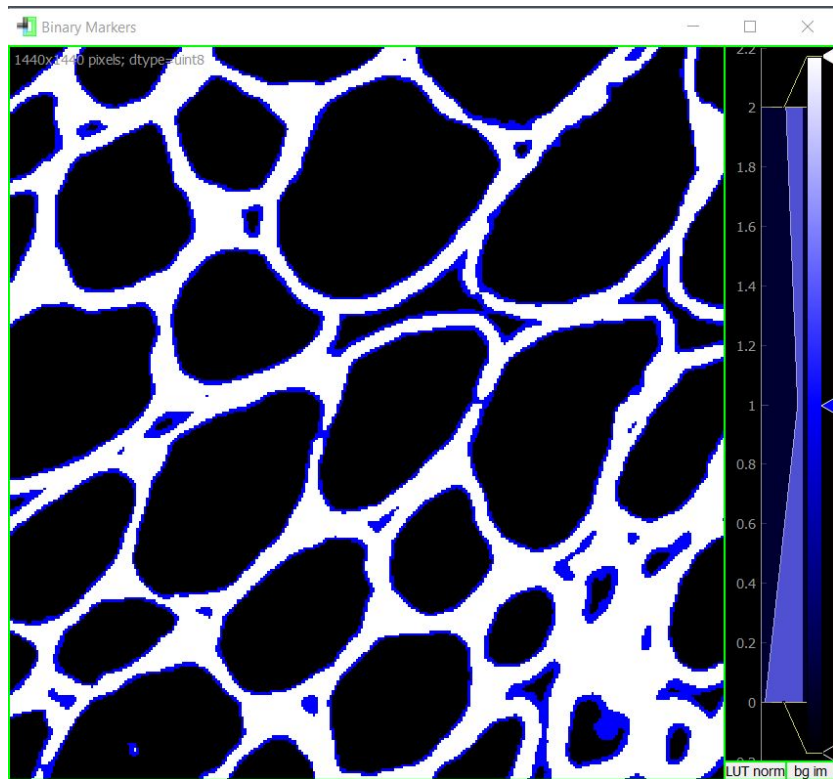
5. To begin the binarization process open the “*Fill Myofiber Gaps*” tab. Click on the ‘*Select Window*’ button and then click on the image of interest (raw laminin image). After selecting your image, a new Flika window tab will appear: “**Binary Markers.**”

Note: To ensure that QuantiMus’s algorithms are being run on the correct image, the user should check that their image of interest is surrounded by a light green outline.





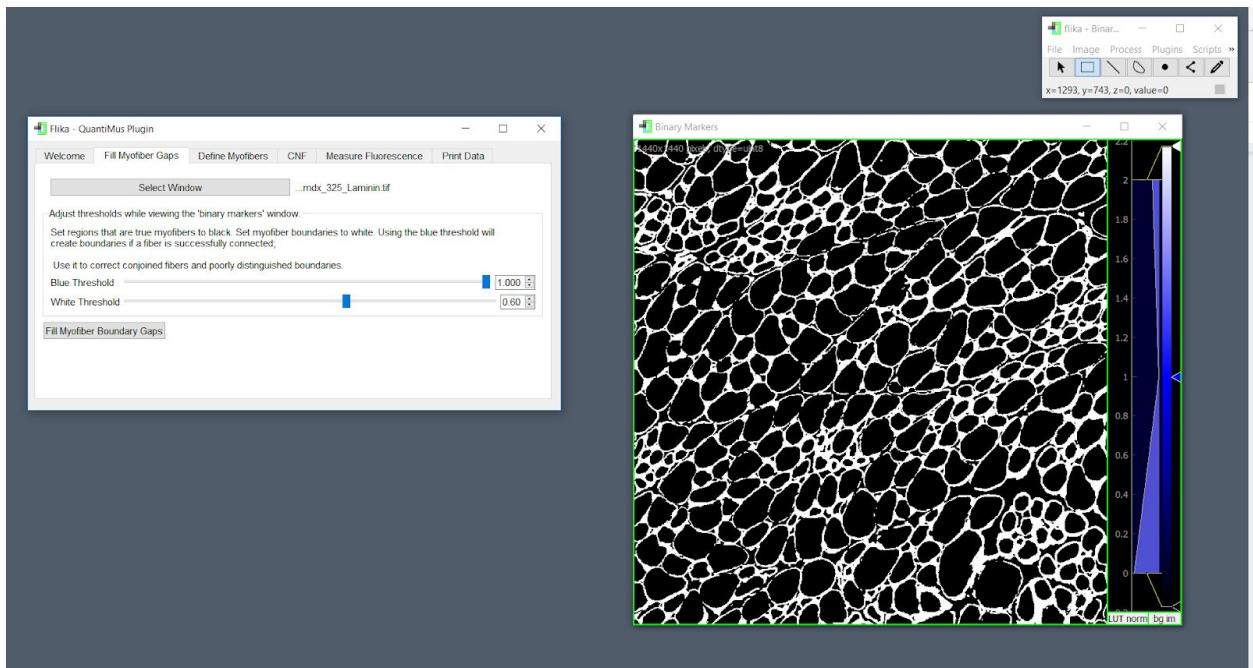
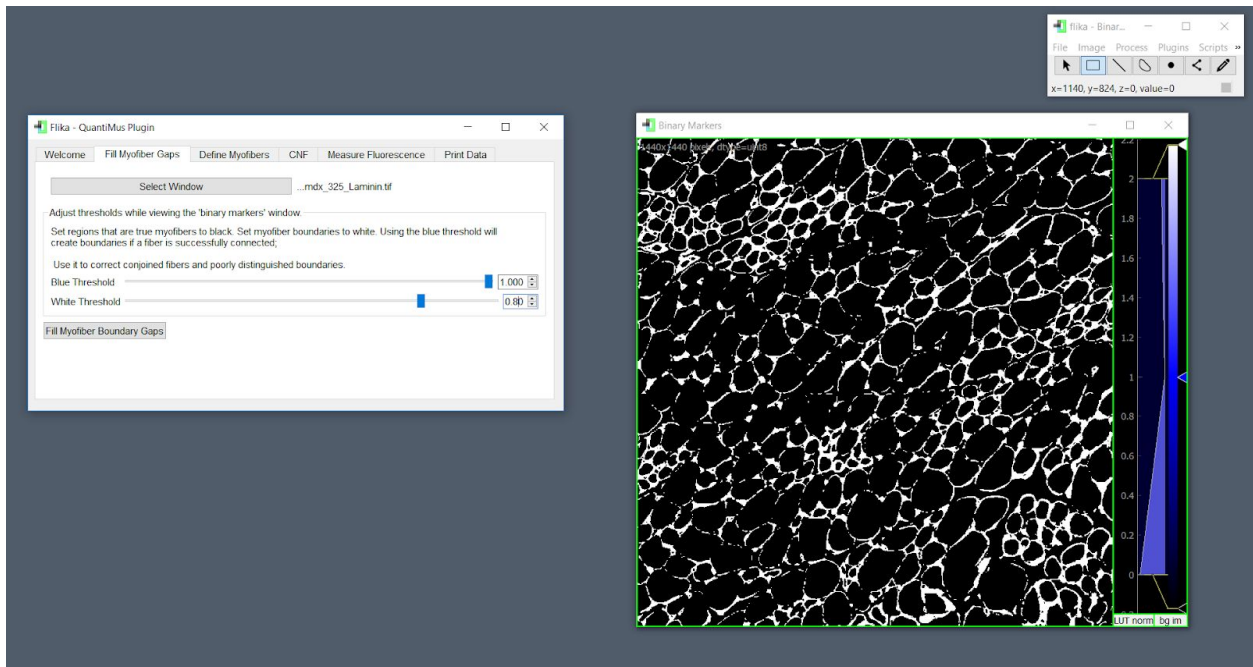
6. The user should now focus on the **“Binary Markers”** window. This where you will begin moving your sliders. Set both of your threshold values so that the blue fiber boundaries are more constricting than the white. All white pixels in this image will become myofiber boundaries in the following step. Blue pixels will become boundaries if their placement restores damaged myofiber boundaries (example below). The user should aim to use the blue threshold to connect white boundaries.

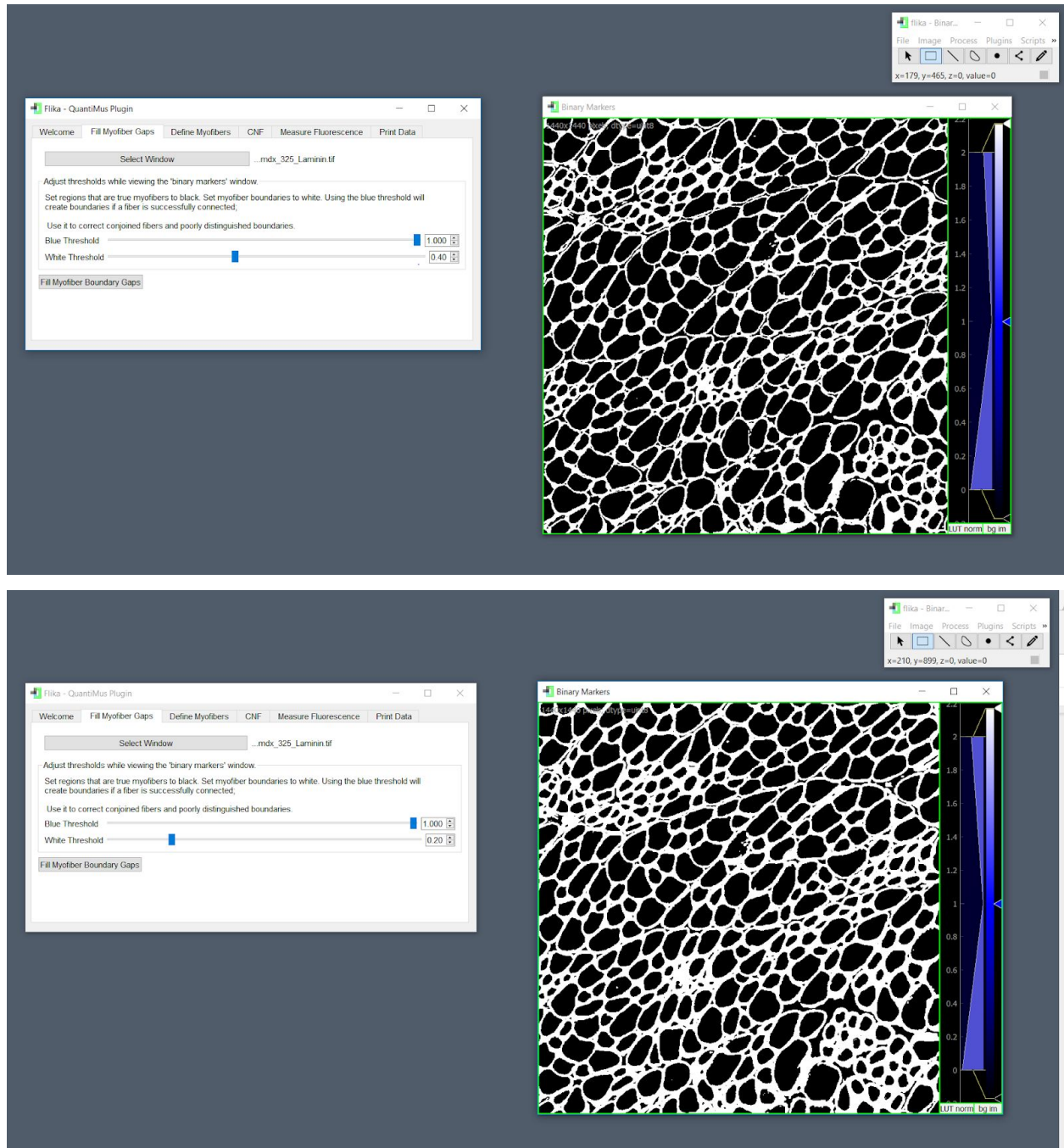


Pockets of entirely blue pixels which represent interstitial space (as shown below) will also be filled.



As a general guide, new users may want to start by setting their threshold values to **1.00**; the result of this will show a blank image. The user can then begin to reduce the value of the '*White Threshold*'. As shown in below in the series of images, Reducing the value of the '*White Threshold*' using the slider will thicken the myofiber boundaries and increase their definition.

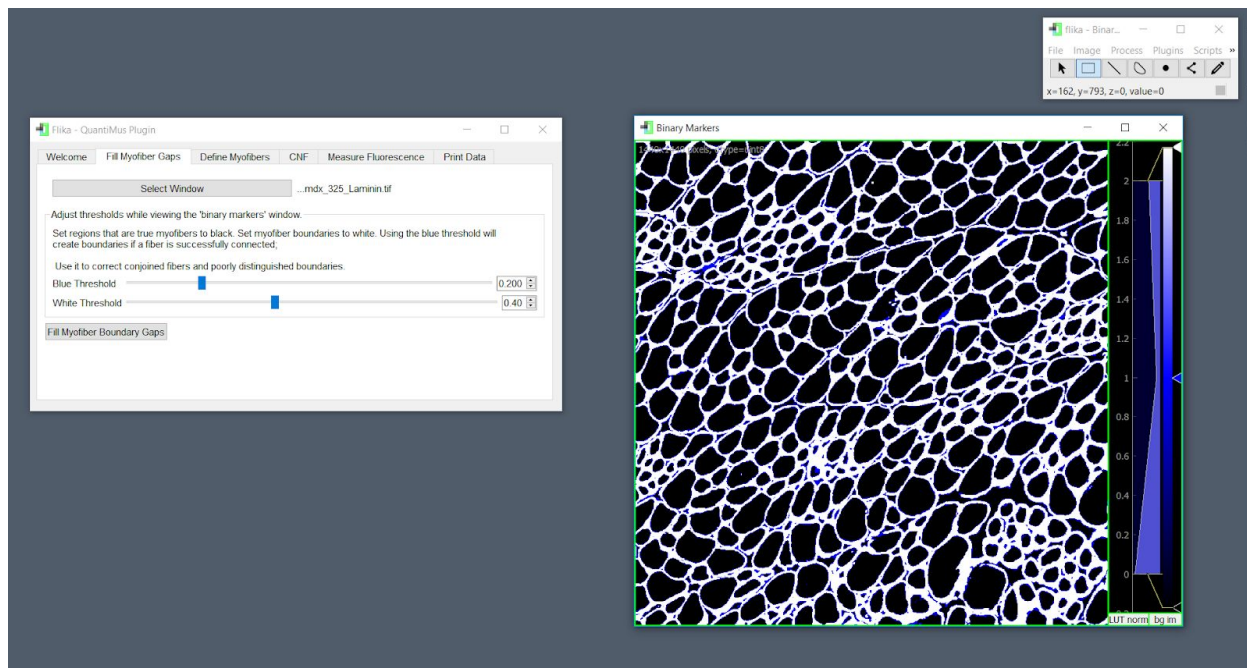




As shown above, increasing the White Threshold to 0.20 severely constricts myofibers. This an important aspect to note because constricted myofibers will inaccurately decrease the area of the myofibers; they will also make smaller myofibers much harder to quantify. In this case, a '*White Threshold*' of 0.4 is deemed appropriate; it is thick enough clearly define myofibers, yet thin enough that it does not hurt myofiber sizes.

After defining an appropriate '*White Threshold*', the user may begin adjusting their '*Blue Threshold*'. The blue threshold may be used more aggressively than the white, because not all blue pixels become myofiber borders. However, an extremely low '*Blue Threshold*' will also negatively affect myofibers. The image below shows the optimal use of the '*Blue Threshold*,' which was 0.200 in this case.

It is important to note that the user should make note of these values for future analysis and experiment normalization. Values may differ depending on specific muscle sections and staining protocols, however using the '*Blue Threshold*' as 0.200 and '*White Threshold*' at 0.40 has proven effective for nearly all our histological analysis in Villalta Lab.

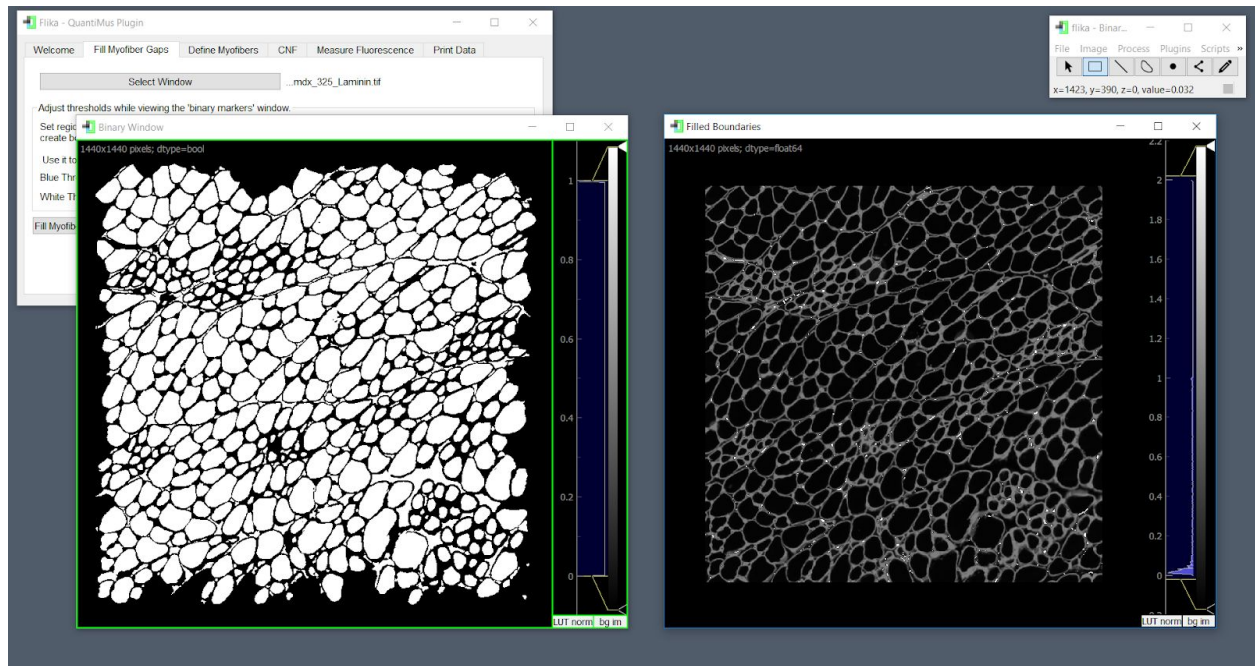


Once appropriate values have been selected, the user should click the '*Fill Myofiber Boundary Gaps*' button. After doing so, QuantiMus will run an algorithm to create your binary image. We have found that *mdx* muscle sections as well as damaged sections take longer to run than uninjured, wild type mice. This algorithm usually requires a couple minutes before a binary image is generated (for a full section 4 wk *mdx* mouse quadriceps).

The images below will then be generated. The "**Binary Window**" will contain the binary image product. A window labeled "**Filled Boundaries**" will also appear, showing the

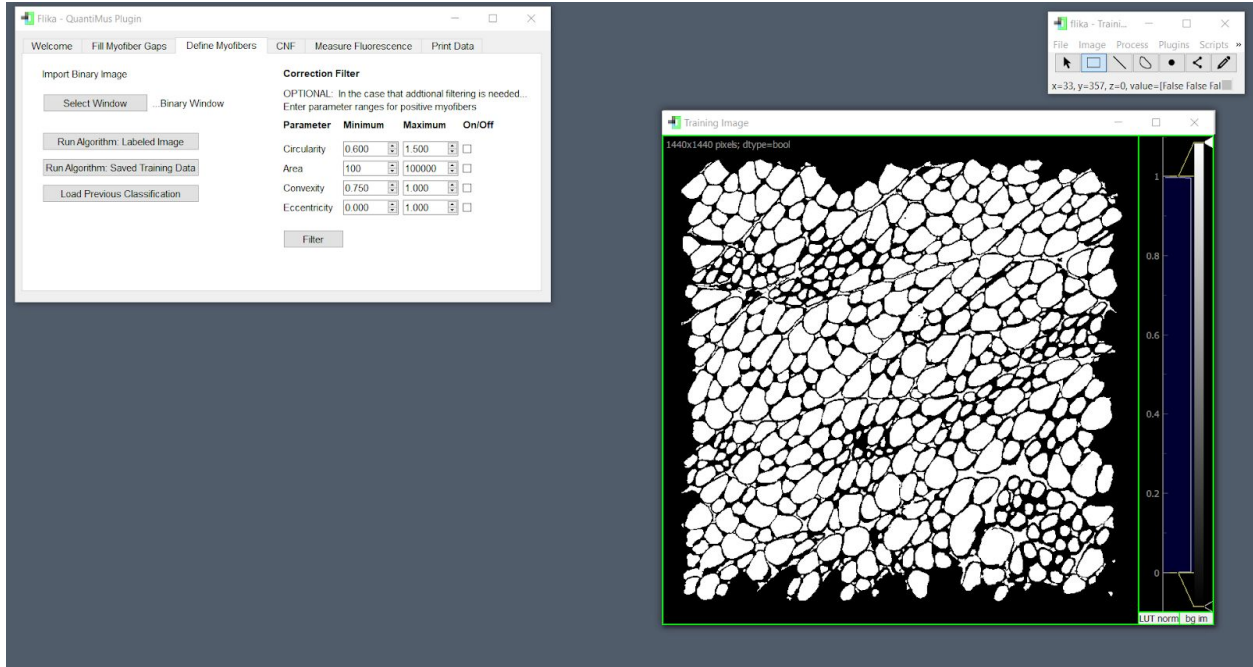
user which parts of the muscle section the “*Fill fiber*” algorithm corrected. The resulting binary image may be saved using Flika: *file -> save -> save image*

Note: Because the user is able to save images throughout the analysis pipeline, the user can create checkpoints for themselves, which they can return to at a later time by opening their previously saved images.



7. After a binary image has been generated, The user may begin to classify the regions of interest as real or fake fibers. The user should click on the ‘*Select Window*’ button then click on their binary image. After doing so, a new window will appear titled:

“Training Image.”



While this window initially looks identical to the last, the **“Training Image”** window is interactive, and will respond to user clicks. Fibers will change colors when clicked to be either red, green, or white. **Green** ROIs will ultimately be classified as real fibers, **Red** ROIs will be classified as artifact or false fibers, and **White** fibers will be ignored. Only **Green** fiber data will be saved for export. The user may do a number of things at this stage in analysis:

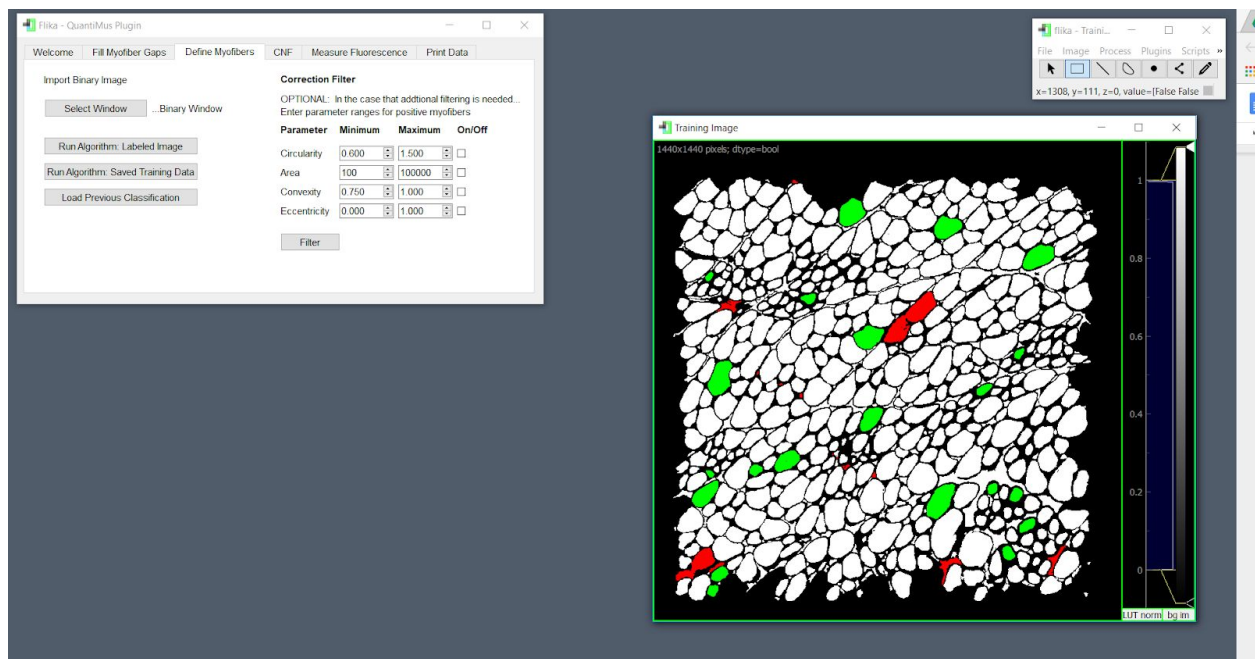
1. **Manually classify their image:** the user may choose to click on ROIs manually to define which are real fibers (**Green**)
2. **Classify their image through labeled fibers:** the user can classify a subset of ROIs (**Green & Red**) and then train the rest of their section using data from those fibers
3. **Classify their image using training data:** the user can choose to classify their image using previously saved training data

Note: Classifying fibers using the labeled fibers (2) or saved training data (3) method uses the same algorithm. Using the saved training data method is often advantageous because it does not require the user to manually classify or manually train their image by hand clicking fibers.

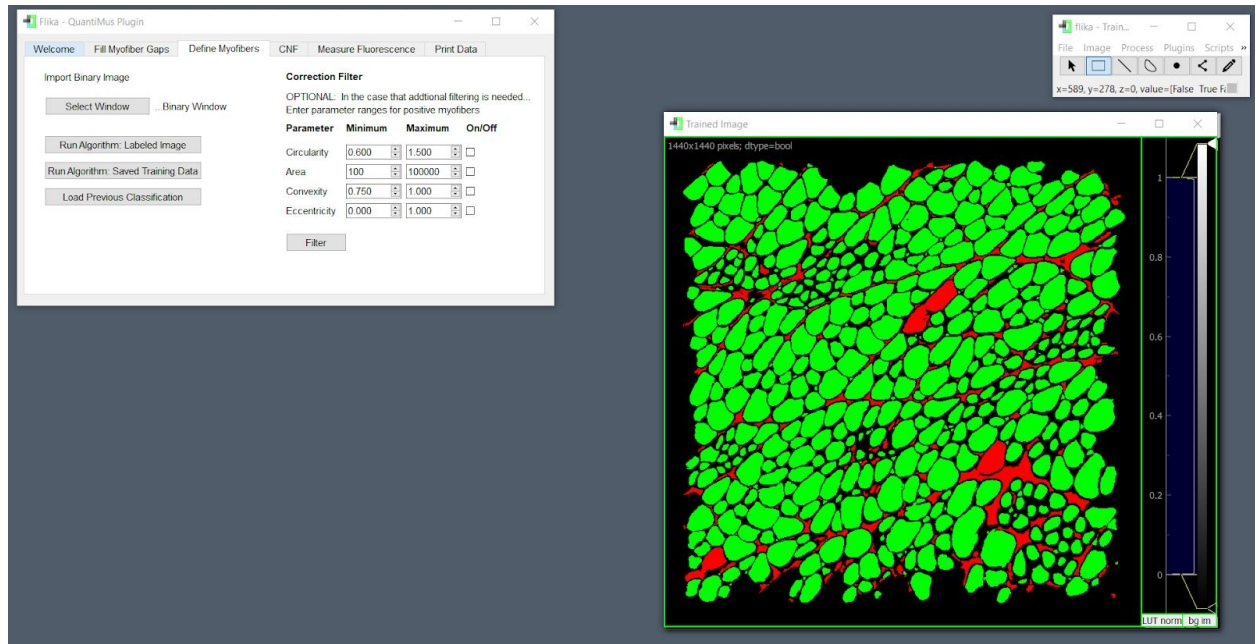
An overview on the SVM machine learning algorithm:

This algorithm is very powerful because it will use a small subset as fibers as an example to classify the remaining fibers. This algorithm thus requires the user to give it examples of real and false myofibers. The algorithm will use geometric properties to distinguish between what the user has selected as a real or false fiber. When selecting fibers (training data) to classify the rest of the image, the user should make sure to click a diverse set of both true and false fibers. Doing so will allow the algorithm to classify fibers of many different shapes and sizes and help reduce false positives and negatives.

For example, if the user only defines large, circular regions of interests as real myofibers, the algorithm will fail to detect small fibers. This is specifically important for full sections and sections from *mdx* mice due to their large diversity in fiber shapes and sizes. An example of a diverse set of fibers is shown below.



Demonstrating method **(2)**: the user clicks the '*Run Algorithm: Labeled Image*' button to run the SVM algorithm. The result is shown below:



This image will be displayed in the ‘*Trained Image*’ Window, which is also user-interactive. After arriving at this stage, the user may choose to use the **Correction filter** in the ‘*Define Myofibers*’ tab (**highly recommended**). The correction filter can be used as an extra step to exclude regions of interest that were misclassified. We highly recommend the user to filter using the “Circularity” and “Area” parameters; abnormally small or irregularly shaped fibers can easily be excluded by this technique.

The user may also choose to manually check the section classifications to change the identity of regions of interest that the algorithm misclassified. After a satisfactory classified image is attained in this step, the user can save both **classifications** and **training data**. These may both be saved by right-clicking the classified image and selecting what the user desires.

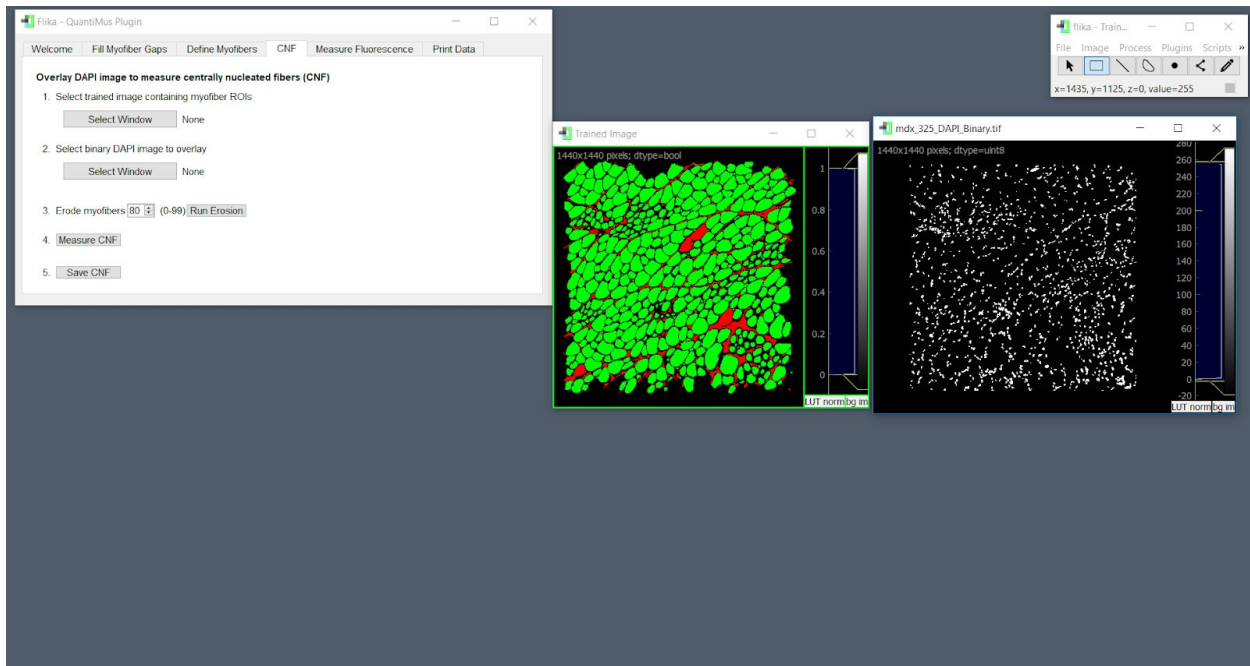
Note: If the classified image is far from complete, we recommend that the user classifies more fibers in their “**Training Image**” window and re-runs their algorithm before filtering and adjusting manually.

Classifications: Saving classifications yields a **.json** file. This file may be used alongside its corresponding binary image to re-obtain the classified image that was used for analysis. The user may also choose to save classifications at the previous step; this way they can return to a “checkpoint” as described earlier (using the ‘*Load Previous Classifications*’ button). We highly recommend that researchers save the

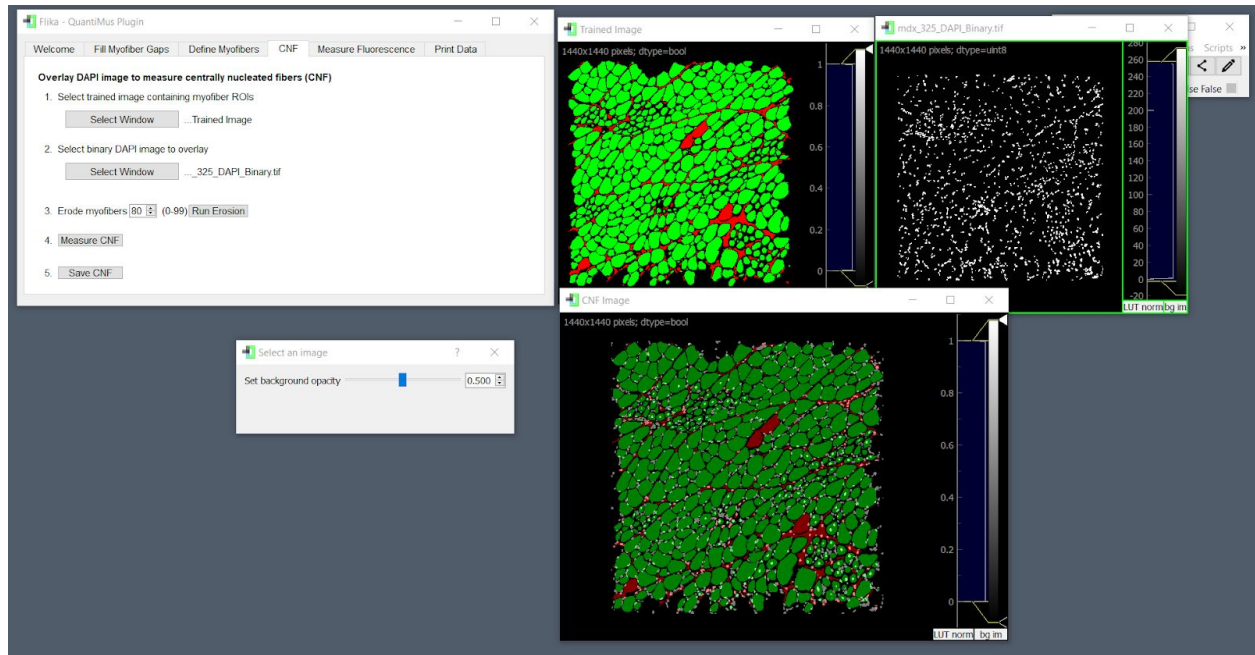
classifications (alongside the binary images used) of their final products. By doing so, the researcher will always be able to re-obtain and re-create these images analyzed.

Training data: Saving training data also yields a **.json** file. This file is different from classifications because it can be used with any binary image. The *'Run Algorithm: Saved Training Data'* button can be used when classifying binary images to find and select these saved files. It is important to note that training data will be based on raw sizes (pixels) and not microns. Thus, training data should be saved and used for specific microscope/pixel ratios.

8. After a satisfactory classified image has been generated in the previous step, the user may choose to search for centrally nucleated fibers (CNFs). The user is required to keep their **'Trained Image'** window open. They are also required to open a binary DAPI-stained image.

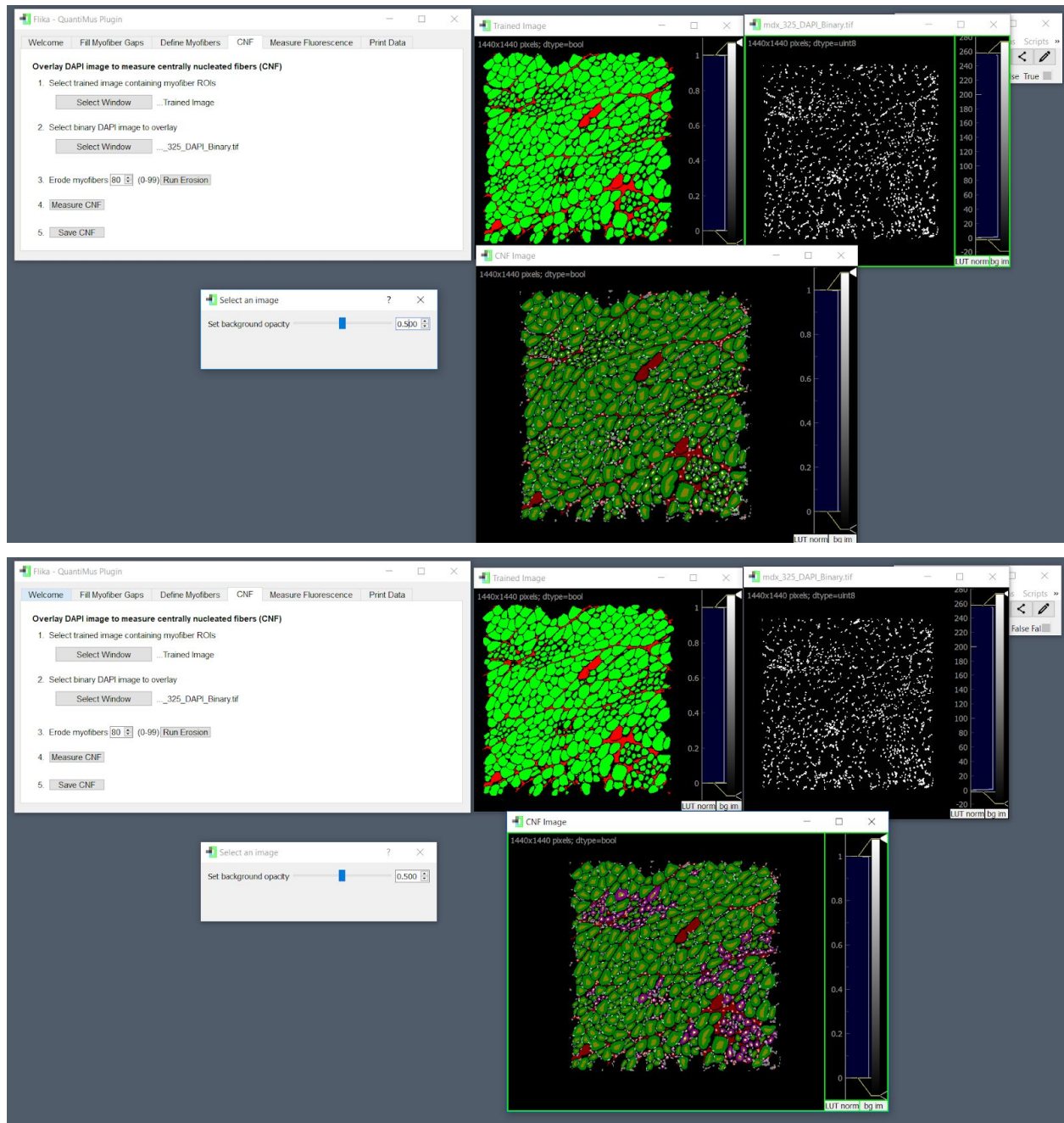


The user then will use the *'Select Window'* buttons to designate these two images in the user interface. After image designation a **'CNF Image'** window will appear. This window displays an overlaid image of the two previously designated images. The user may adjust the opacity of this overlay using the slider. This is purely aesthetic and will have no effect on the downstream analysis.



After reaching this image, the user can enter a number in the user interface that will dictate how which fibers are classified as CNFs. A number between 0 and 99 should be entered, with 99 being the most strict. The eroded fibers will be colored yellow after the erosion algorithm has been run. After erosion, the user should select the '*Measure CNF*' button to display CNFs, which will be colored purple.

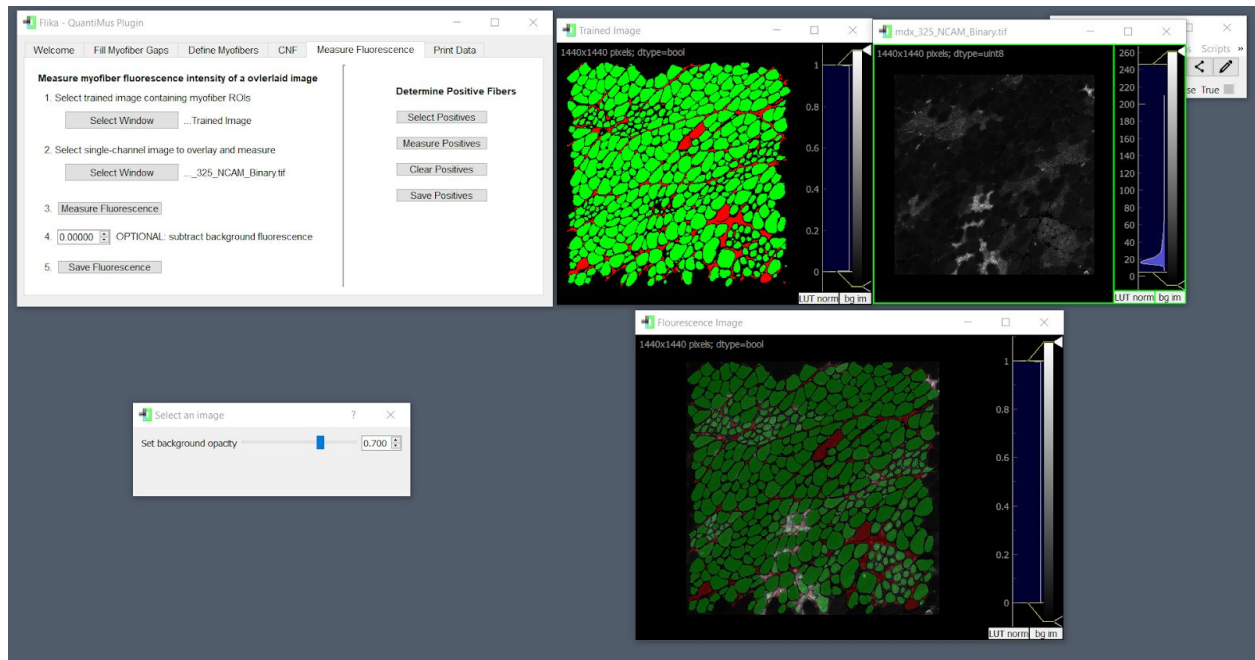
Note: a value of "80" means that 80 percent of the myofiber will be eroded down to its center. If a nuclei is present in this eroded location, the fiber will be classified as a CNF.



As with previous classifications, the user may click on fibers to override their status to correct any errors (they may also save them). Once the user is happy with their image, they should click the 'Save *CNF*' button to save the identify of CNFs for downstream analysis.

8. To measure the MFI (mean fluorescence intensity) of individual fibers, the user should have their **Trained Image** window open, as well as an 8-bit image of their

fluorescence stained section. For this example, the fluorescent stain of interest was done with neural cell adhesion molecule (NCAM). As done before, the user should designate images on the user interface. Once this is done, a **‘Fluorescence Image’** window will appear.



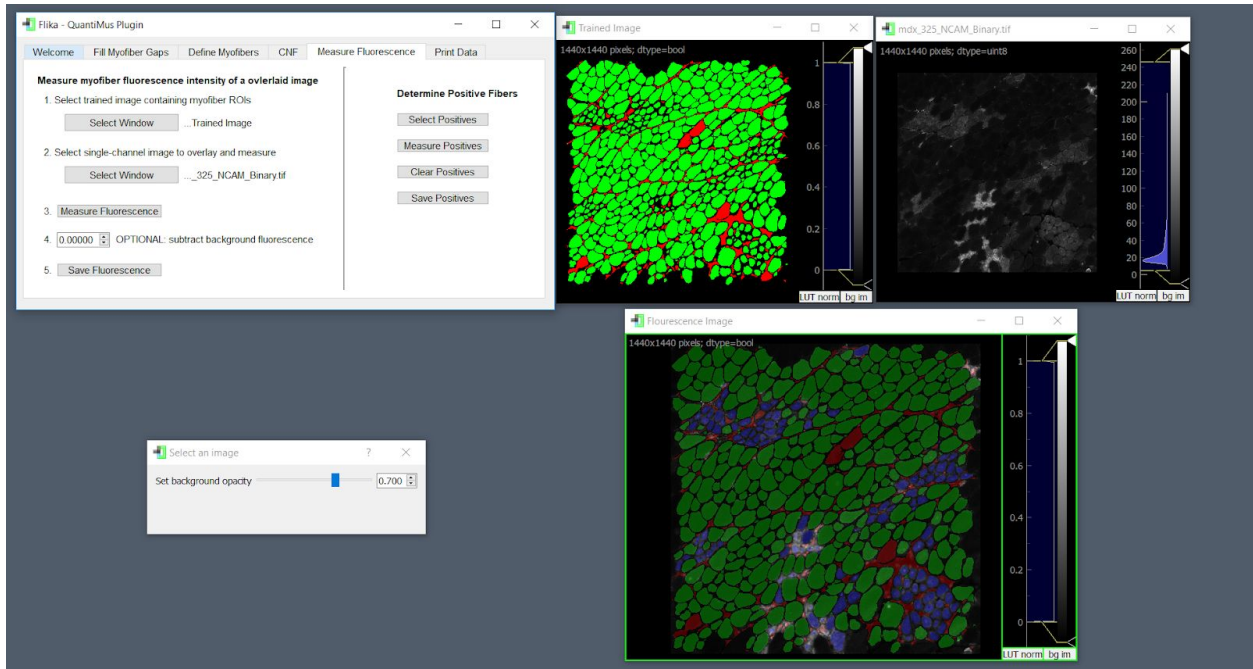
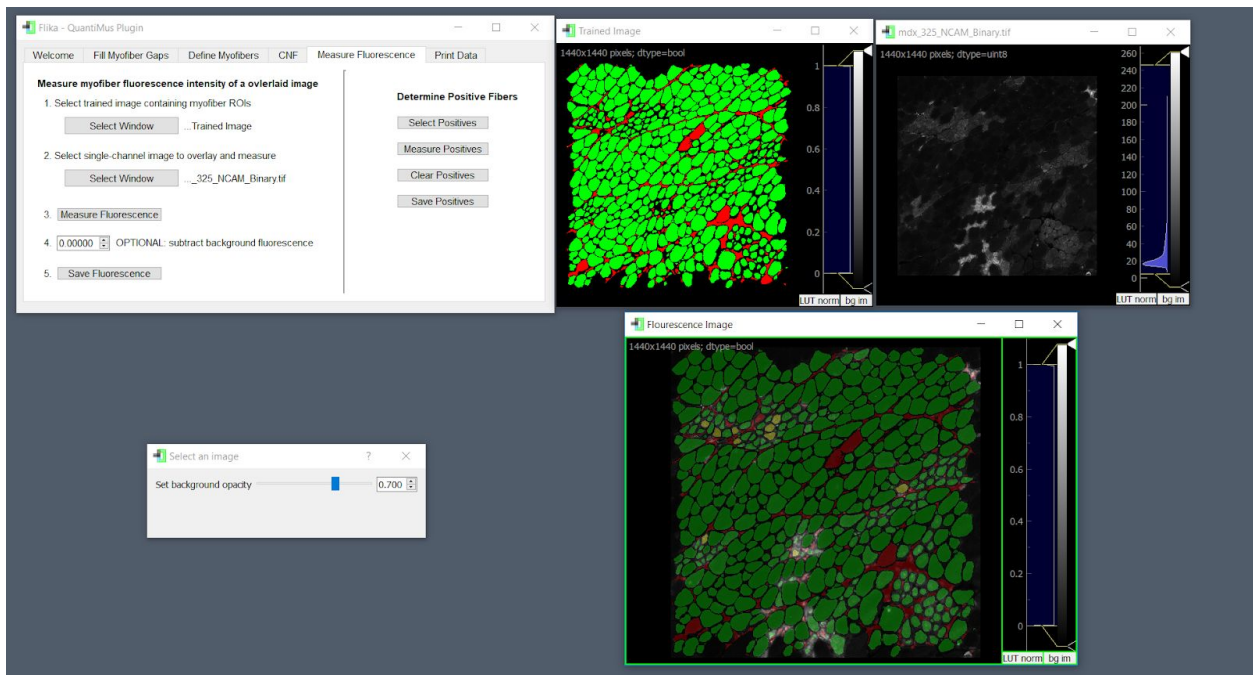
The user then clicks the *‘Measure Fluorescence’* button on the left side of the UI to measure the MFI. The user also has the optional choice of subtracting background fluorescence by entering a value. This will subtract that value from the MFI value of each fiber; this is specifically useful to subtract background fluorescence captured during image acquisition. Once measured, the user should use the *‘Save Fluorescence’* button.

Note: MFIs of ROIs may be inspected by simply clicking on them. The MFI for that specific ROI will be displayed in the terminal.

The user also has the option of determining positive fibers (on the right side of the UI). Using this feature, the user can identify which myofibers are positive for expression of the protein of interest.

First, the user clicks the *‘Select Positives.’* After doing so, the **‘Fluorescence Image’** will become interactive. Double clicking fibers will turn them yellow. The user should select fibers that are just fluorescent enough to be classified as positive for the marker.

The lowest MFI value will be extracted from the yellow fibers, and this value will serve as the threshold number that dictates if fibers are positive or not.

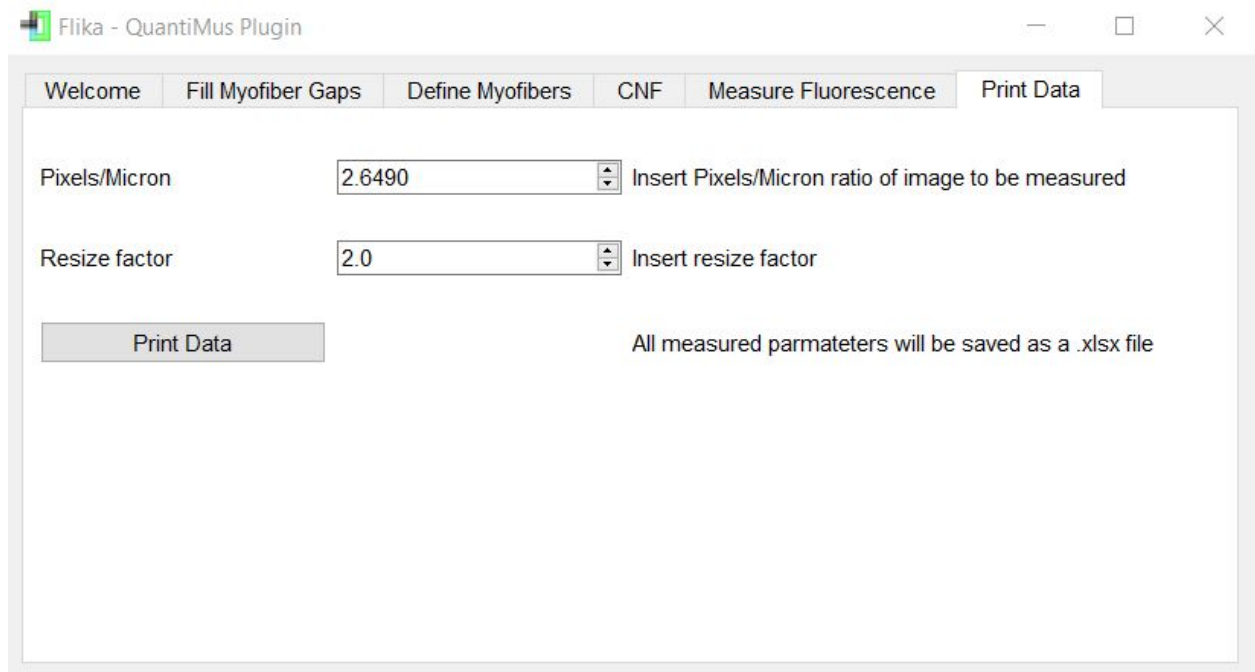


After selecting yellow fibers, the user can click on the '*Measure Positives*' button to turn all positive fibers blue. These blue fibers represent positive fibers. The results of this

process may be discarded to allow the user to try again (*'Clear Positives'* button) or saved (*'Save Positives'*).

9. Once the user has completed analysis, they can enter their respective Pixel/Micron ratio as well as a resize factor, if relevant. A resize factor of 2 indicates that the length and width of the raw image have been multiplied by 2. Thus, an image that is resized by a factor of 2 will contain 4 times as many fibers. Most users will not resize their images, and thus enter a resize factor of 1.

Note: We generally do not recommend resizing images, as this increases processing time.



The screenshot shows the 'Flika - QuantiMus Plugin' window with the 'Print Data' tab selected. The window has a title bar with standard Windows controls. Below the title bar is a tabbed interface with the following tabs: 'Welcome', 'Fill Myofiber Gaps', 'Define Myofibers', 'CNF', 'Measure Fluorescence', and 'Print Data'. The 'Print Data' tab is active and contains two input fields with spinners. The first field is labeled 'Pixels/Micron' and has the value '2.6490', with a placeholder text 'Insert Pixels/Micron ratio of image to be measured'. The second field is labeled 'Resize factor' and has the value '2.0', with a placeholder text 'Insert resize factor'. Below these fields is a 'Print Data' button. To the right of the button, a message states: 'All measured parameters will be saved as a .xlsx file'.

The user can then *'Print Data'* to save their results as a Microsoft Excel file.

Note: CNF fibers will be identified with a number "1"

	A	B	C	D	E	F	G	H	I
	ROI #	Area	Minferet	CNF	MFI	Positive	Scale Factor	Resize Factor	
2	2	138.0536	10.16062		13.37316	0	2.649	2	
3	4	80.76582	9.72592		15.98765	0			
4	5	0.035627	0.377501		23	0			
5	8	36.80242	5.030234		24.73379	0			
6	9	63.73624	7.527454		16.35998	0			
7	10	182.7652	13.75368		15.21014	0			
8	11	41.86142	6.670188		24.48851	0			
9	12	71.00409	7.819548		18.05569	0			
0	13	30.42524	4.270804		42.69204	1			
1	14	45.99412	7.764258		18.64911	0			
2	16	189.6411	10.89196		14.90663	0			
3	17	60.13794	8.100095		18.34953	0			
4	18	125.5486	10.56557		13.94864	0			
5	19	178.4187	11.32247		19.95327	0			
6	21	169.2626	11.89782		17.43675	0			
7	22	178.3475	11.23244		29.33899	0			
8	23	56.14774	6.676951		16.78426	0			
9	25	33.02599	4.809055		32.57066	0			
0	26	122.6272	11.97327		18.79895	0			
1	27	51.05312	6.723338		18.26867	0			
2	28	146.2478	11.7659		16.77467	0			
3	31	104.0657	8.546694		22.56761	0			
4	34	96.47721	8.57894		19.83973	0			
5	35	151.6274	11.13743		14.60691	0			