Documentation

Overview:

This program is designed to analyze and identify anisotropic cell clusters along with isotropic cell clusters.

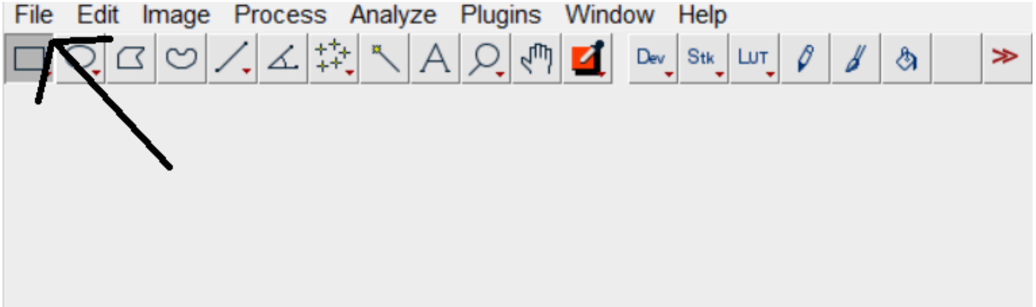
Setup:

There are a few key rules to follow while setting up images to analyze. The program is designed to identify images up to a certain brightness; images that are too bright or not bright enough will produce errors and unwanted results.

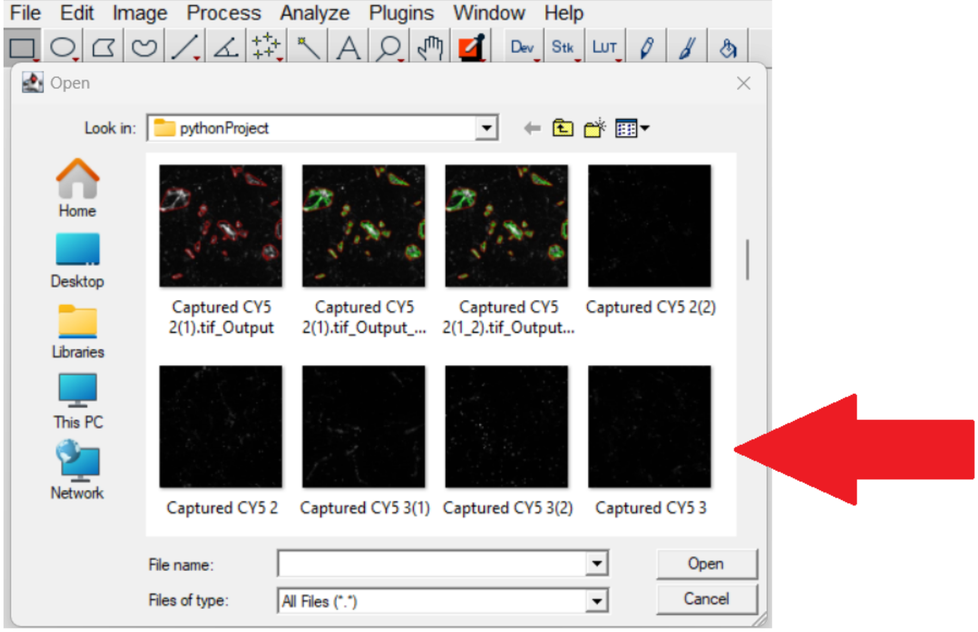
Brightness(Figi Image):

Softwares such as Fiji Image or Wondershare are great ways to enhance brightness and visibility. For example, setting the brightness in Fiji Image to optimal levels is as followed:

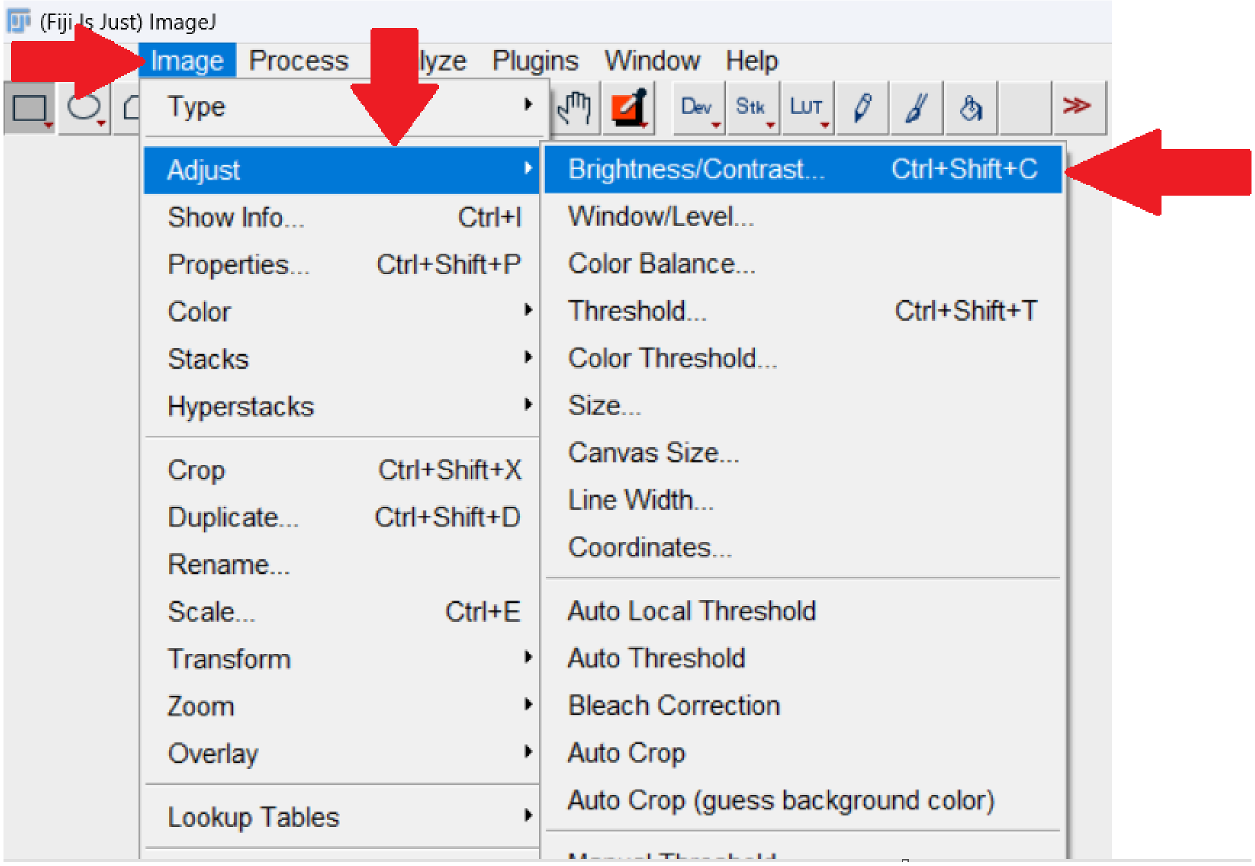
1. Head over to File → Open



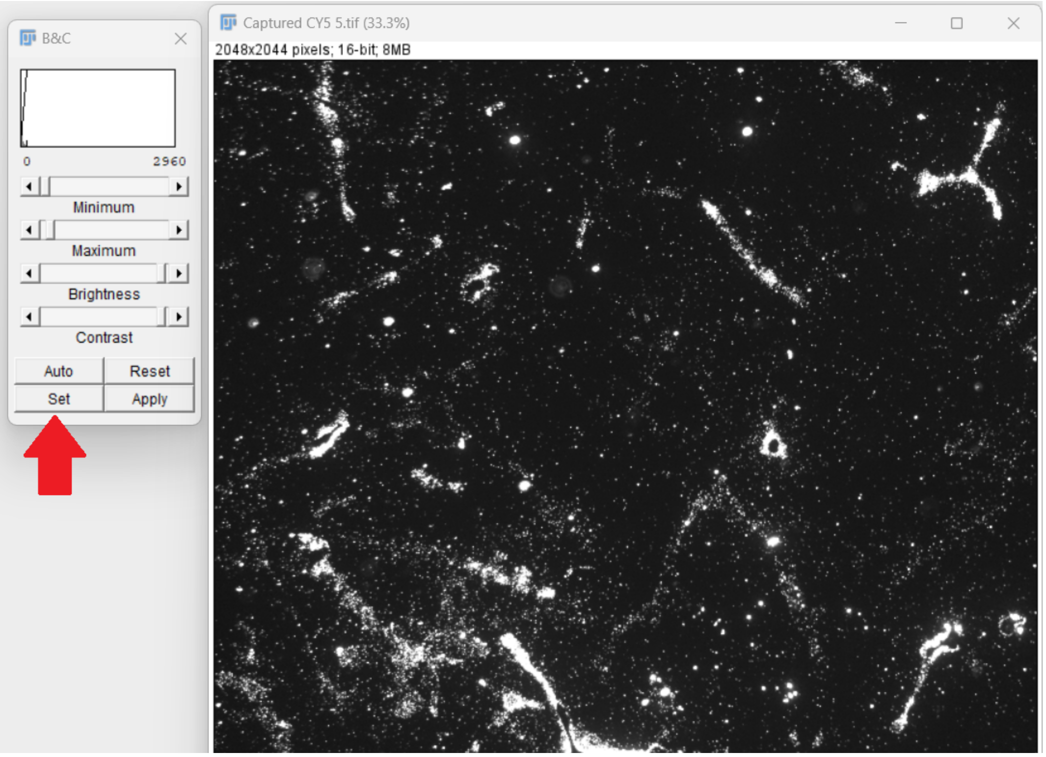
2. Select your desired Image and click Open.


File Edit Image Process Analyze Plugins Window Help
Dev Stk LUT
Open
Look in: pythonProject
Home
Desktop
Libraries
This PC
Network
Captured CY5 2(1).tif_Output
Captured CY5 2(1).tif_Output_...
Captured CY5 2(1_2).tif_Output...
Captured CY5 2(2)
Captured CY5 2
Captured CY5 3(1)
Captured CY5 3(2)
Captured CY5 3
File name:
Files of type: All Files (*.*)
Open
Cancel

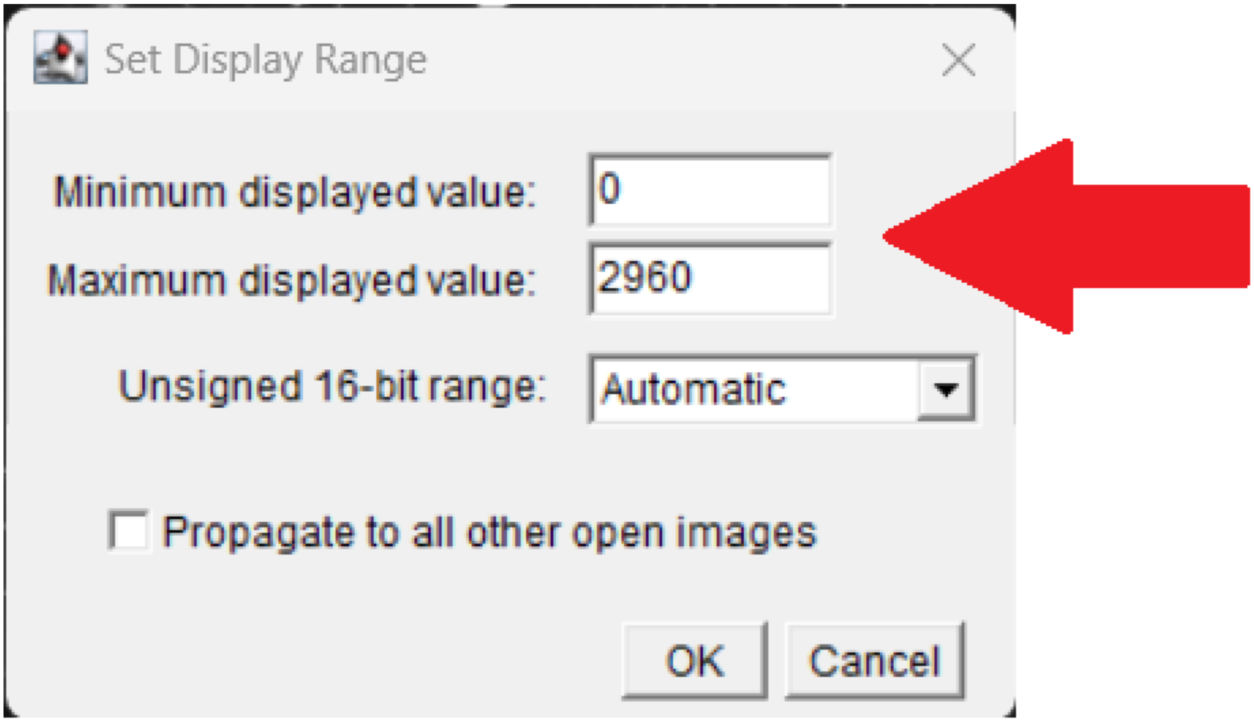
3. Once your Image loads, click on Image → Adjust → Brightness/Contrast



4. Click on “Set” once the “B&C” window opens.



5. Once the “Set Display Range” window opens. Set the values to the ones shown in the image below.



Set Display Range

Minimum displayed value: 0

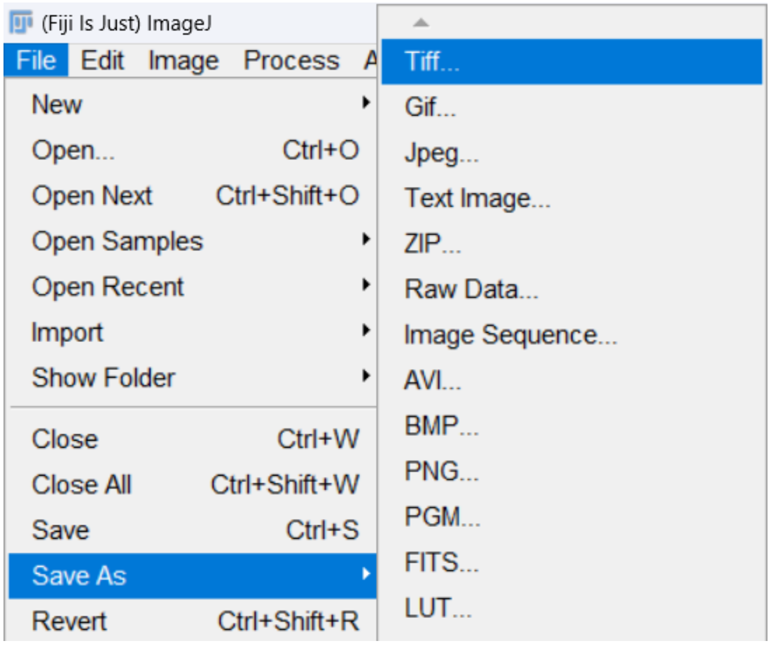
Maximum displayed value: 2960

Unsigned 16-bit range: Automatic

□ Propagate to all other open images

Note: The values shown above are the recommended values for an image. As long as the minimum displayed value doesn't exceed 1000 and the maximum value is above 5500, the program will work as intended. (1000<x<5500)

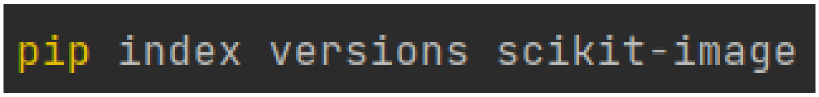
6. Once the values are set and applied, go ahead and save the image as your desired file name.



Program Setup:

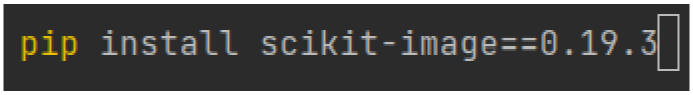
The program is quite easy to set up although it's essential to have to right versions of the packages in order to run the program.

1. Skikit-image version 0.20.0 or higher is not compatible with the Watershed algorithm with prevents the program from running. In order to ensure your Skikit-image is below the version requirements run the following command below in your terminal:



```
pip index versions scikit-image
```

2. To install a certain version of a library run the following command:

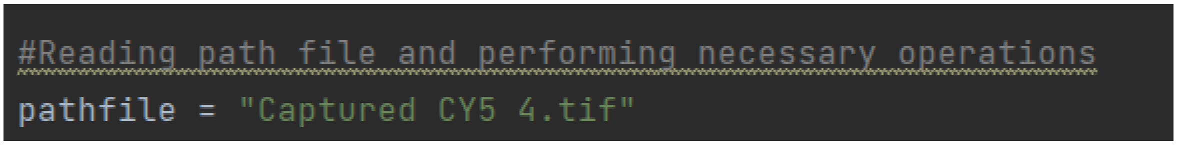


```
pip install scikit-image==0.19.3
```

Running the Program:

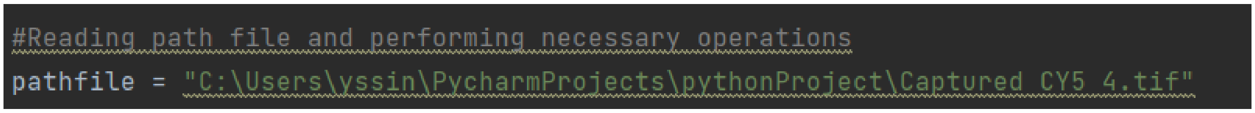
There are a few user customizations available within the program. The following guide will list all of the customization options which may be utilized for certain images.

1. The easiest way to select an image to check for clusters would be to either change the value of the variable "pathfile" to either the path or the name of the image file. The picture above shows an example of the following.



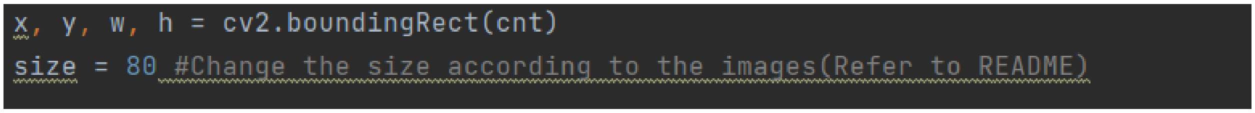
```
#Reading path file and performing necessary operations
pathfile = "Captured CY5 4.tif"
```

Or



```
#Reading path file and performing necessary operations
pathfile = "C:\Users\yssin\PycharmProjects\pythonProject\Captured CY5 4.tif"
```

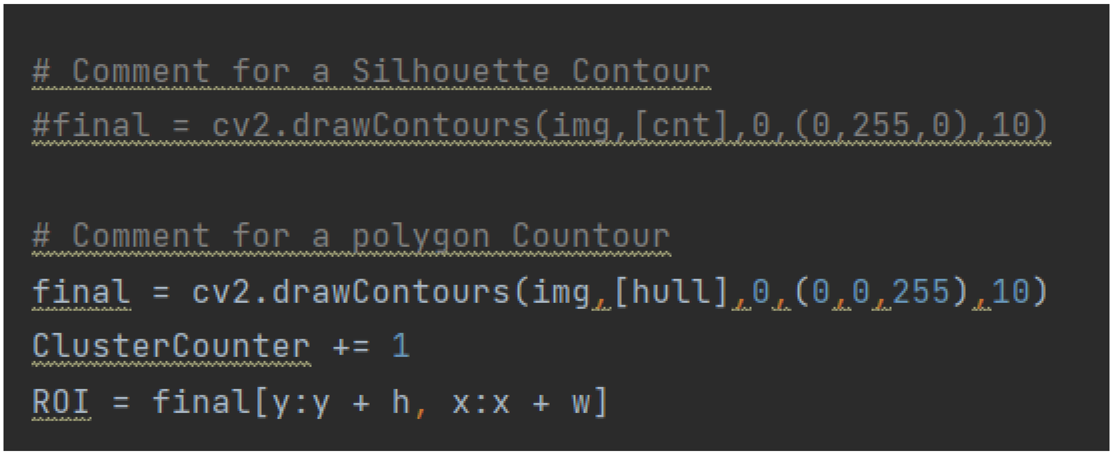
2. Another feature is the ability to change the minimum size a cluster must be in order to get recognized. The following image shows how a user is able to customize the size.



```
x, y, w, h = cv2.boundingRect(cnt)
size = 80 #Change the size according to the images(Refer to README)
```

****Note: The recommended value for most images is size 80; however, some images may require a higher or lower size value.****

3. The final customization parameter is the "draw. countours" parameter. This program has two types of contours built-in: the first is a silhouette contour which draws a silhouette of the cluster and the polygon contour which draws a polygon around the identified cluster. The image shown below contains more details about the contour parameter.



```
# Comment for a Silhouette Contour
#final = cv2.drawContours(img,[cnt],0,(0,255,0),10)

# Comment for a polygon Countour
final = cv2.drawContours(img,[hull],0,(0,0,255),10)
ClusterCounter += 1
ROI = final[y:y + h, x:x + w]
```

Final points:

Overall, each image has its own properties and is unique in its own way. It's important to tinker with the parameters to get optimal results. The images shown below are a few of the final outputs depicted by the program.

Cell Cluster Images.