NPC Image Analysis

Release 0.1

Ashley Hardy

May 26, 2022

CONTENTS

NPC Image Analysis is a Python app for analyzing 3D images of Neural Progenitor Cells (NPCs). In particular, it was made to answer the specific question of analyzing the radial process of the cells. This application aims to allow you to use a default implementation to get you started, but also allows the user to customize the analysis process.

Note: This project is still under active development.

CHAPTER

ONE

WHY NPC IMAGE ANALYSIS?

Originally, this project was aimed to answer a specific question by looking at NPCs in developing zebrafish brains. The purpose of this, what to hopefully eliminate some of the biase that can come with trying to perform image analysis. As the sole developer of this project, I grew attached to continuously iterating on this project. While working on it and trying to explain this image analysis project to my PI, I found that the growing Python scripts might be overwhelming for a non-programmer. Thus, this project has been born to hopefully bridge the gap between trying to analyze the images in a specific context.

This may have other purposes beyond this specific context, so I decided to share it here.

Check out the *Usage* section for further information.

You can find the source code here

CHAPTER

TWO

NAVIGATION

2.1 Usage

2.1.1 Requirements

Python 3.6 or above

2.1.2 Installation

```
pip install npc-analyze-image
```

However, the sourcecode of this program can be found at this github page.

Once installed you can run the following in the command line to initate the program.

```
python -m NPCAnalysis
```

2.1.3 General Usage

This software is intended to be modifiable in order to answer particular needs to identify organelles and structures. Using this software, you will be able to look within the radial process of a specific cell of interest. This is used in conjuction with ImageJ's Simple Neurite Tracer.

To use this software, you will need…

- A greyscale tiff image that contains the channel for the organelles or structures of interest you would like analyzed
- The skeleton file created from the simple neurite tracer as a .tiff file. You cannot use the .trace file for this as it cannot be read in.

After you load in your images using the "Load Images" button, you can selected "Perform Analysis". You will be provided with two options, "Default Analysis" and "Custom Analysis". If you would like to just see what a default analysis would look like, selected Default Analysis.

This will ask for the "z" spacing of your images. This only effects the presentation of your images not the analysis. From there, depending on the size of the image, it can take a while for the analysis to complete. If it looks temporarily frozen, that's possibly why. You can find more information about the Default Analysis option and what operations are performed, selected the Default Analysis under the help window.

You can see examples of this file being used in *Example Usage*

While this program is tailored to a specific use case. The default analysis presumes that you are loading using a skeleton mask (or a skeleton/mask file in general) along with an image to be analyzed. As a result, the program will require you

to load two images and certain operations will presume that both files are loaded. However, you can use most of the operations the skeleton file.

2.1.4 Why use a skeleton mask?

Originally this was used to isolate the file to a specific location for the analysis, however, this also reduced the overall file size and image size lowering the memory usage and time required to process images.

2.2 Default Analysis

The default analysis was created based off an analysis process that worked generally will on all of the images that this project was originally created for.

As such, this process is automatic when you click the button to start the analysis preventing any intervention.

This can be used as a general guide for what analysis processes might work. Image analysis is a process that doesn't work for all images so it can require tweaking to better suit images loaded into the application.

Note: The default analysis presumes that you are using the skeleton file and the image to be analyzed.

2.2.1 Image Processing Steps

- Skeleton Dilation (Default dilation is 10)
- Cropping of the image to the length/size/volume of the skeleton
- Median filter with a cube width of 3
- Background subtraction using a gaussion blur of strength 7.
- Adaptive Histogram Equalization: This consists of using scikit image's adaptive histogram equalization option. It is then multipled to the background subtracted image to give the final contrast.
- Multiotsu mask at 2 classes (there are presumed only 2 classes, background and regions of interest)
- Default morphology: Dilation morphology and then closing morphology
- Labels are added to the image.

This information can also be found in the "Help" section of the application.

2.3 Custom Analysis Options

Custom analysis allows you to customize the image processing steps. This is created to allow the user the opportunity to tailor the processing to the best of their ability without having to worry about the code behind it.

For information on what an operation might do, the scikit image page (which these processing steps draw on) will be linked to allow the user to read more on its purpose.

Below is a general list of the steps.

2.3.1 Main Processing Options

- **Skeleton Dilation**
  - Utilizes the Morphology Binary Dilation Operation
- **Median Filter**
  - Uses the Median Filter from skimage
- **Background Subtraction**
  - Utilizes SciKit Image Gaussian Filter
- **Sobel Filter**
  - Uses the Sobel Filter
- **Adaptive Histogram Equalization**
  - This consists of using scikit image's adaptive histogram equalization option. It is then multipled to the background subtracted image to give the final contrast.
  - Uses the Adaptive Histogram Equalization from skimage
- **Rescale Intensity operation**
  - Uses the Rescale Intensity from skimage

2.3.2 Mask Options

- MultiOtsu Mask
- Otsu Mask
- Yen Mask
- Li Mask

2.3.3 Morphology Options

- **Default Morphology**
  - Includes binary dilation and the closing. Used in the default analysis process
- Binary Closing Morphology
- Binary Opening Morphology
- Binary Dilation Morphology
- Binary Erosion Morphology
- **Get Labels**
  - Obtain the labels after the morphology to get your ROIs

2.3.4 Undo/Redo

Note: Previous image can only go back once.

- Use Previous Image (will use the previously created image prior to the most recent action)
- Use Most Recent Image (this will return you to the current image if you clicked previous image)
- Use Original Image (reset to the originally loaded in image)
- Use Previous Mask (will use the previously created mask prior to the most recent action)
- Use Most Recent Image (this will return you to the current mask if you clicked previous mask)

This information can also be found in the "Help" section of the application.

2.4 Region Properties

2.4.1 area

The property calculates the area of the region of interest. For 3D options this really means volume as it is taking into account the x, y, and z planes.

2.4.2 bbox

Calculates the boundary box points locations for each plane. This returns six columns as it accounts for the highest and lowest x, y, and z planes coordinates.

2.4.3 bbox_area

Calculates the bounding box area based on the bounding box coordinate points. As this is 3D this is closer to volume than area.

2.4.4 max_intensity

Calculates the maximum intensity for each of the ROIs identified. This takes into account the passed in intensity_image. For this program, it uses the original image so that the intensity values are not impacted by the processing steps.

2.4.5 mean_intensity

Calculates the mean intensity for each of the the ROIs identified. This takes into account the passed in intensity_image. For this program, it uses the original image so that the intensity values are not impacted by the processing steps.

2.4.6 min_intensity

Calculates the min intensity for each of the the ROIs identified. This takes into account the passed in intensity_image. For this program, it uses the original image so that the intensity values are not impacted by the processing steps.

2.4.7 equivalent_diameter

Calculates the diameter of a circle with the same area as the region.

2.4.8 minor_axis_length

The length of the minor axis of the ellipse that has the same normalized second central moments as the region; that is, the same normalized variance as the region which is a measure used to quantify whether the set obsevered occurences are clustered or dispersed.

2.4.9 major_axis_length

The length of the major axis of the ellipse that has the same normalized second central moments as the region; that is, the same normalized variance as the region which is a measure used to quantify whether the set obsevered occurences are clustered or dispersed.

2.4.10 centroid

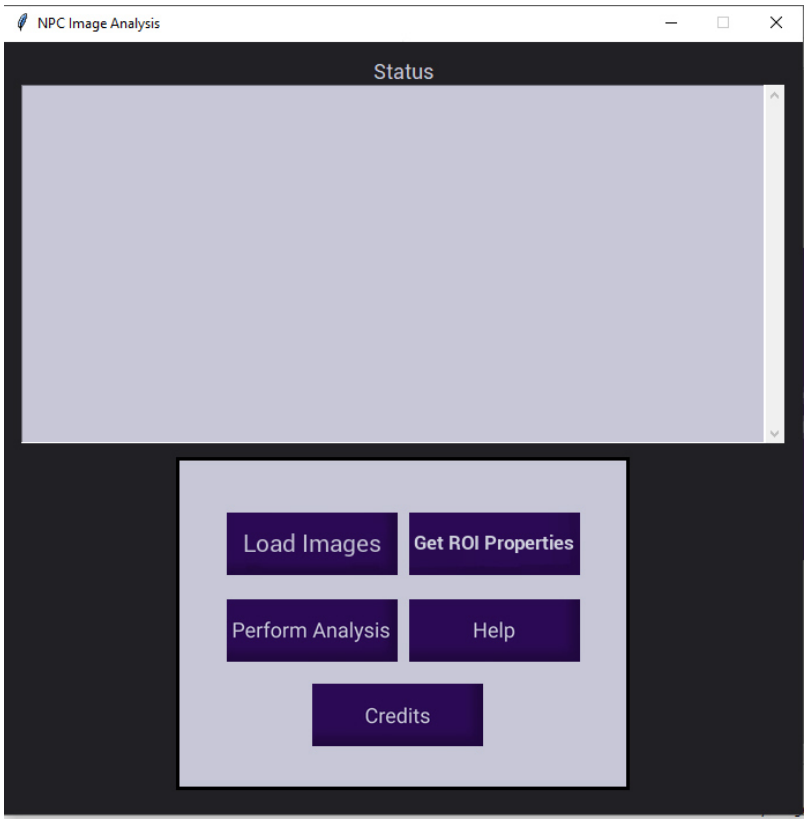
This returns the coordinates of the central point within the specific region of interest. Creates three columns with x, y, and z coordinates.

2.4.11 coords

Returns the coordinate list (plane, row, col) of the region

2.5 Example Usage

2.5.1 Using NPC Image Analysis



Above is what the application will look like upon initialization. The application's simple interface was designed to not overwhelm the user with too much information and options at one time.

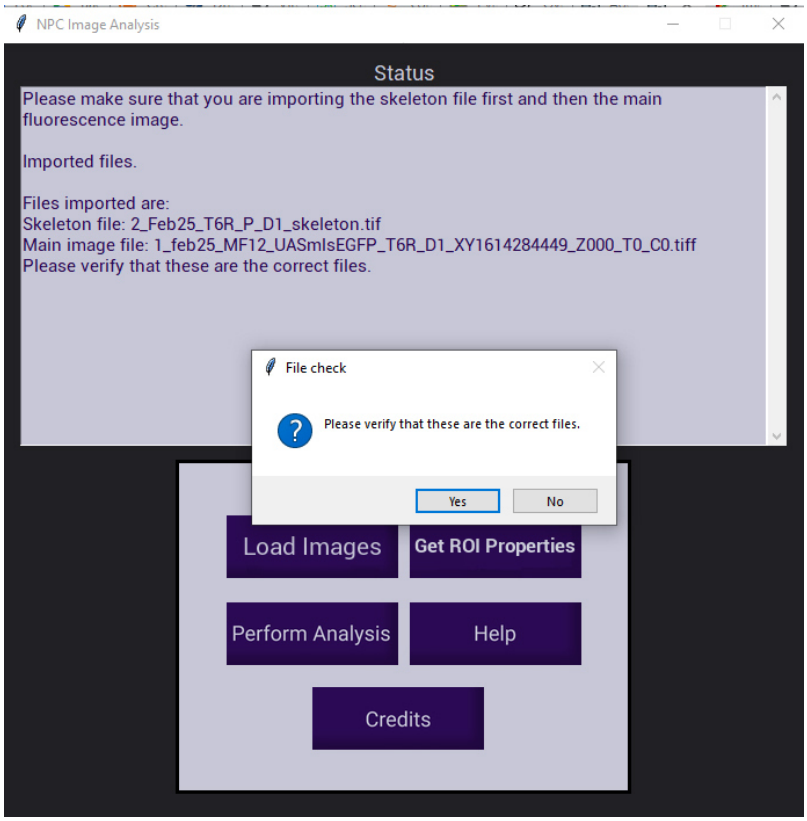
We'll start with loading in the images. When the loading button is clicked, the status screen will inform you of the order of the images.

Warning: You must import the images in a specific order:

1. Skeleton file first (this must be a .tiff or .tif file)
2. Your image for analysis

They must be the same size, or else the application will not be able to process the images.

After the images are loaded, the "Status" screen of the application will list the images loaded and will ask the user to confirm that these are correct.



2.5.2 Choosing an analysis option

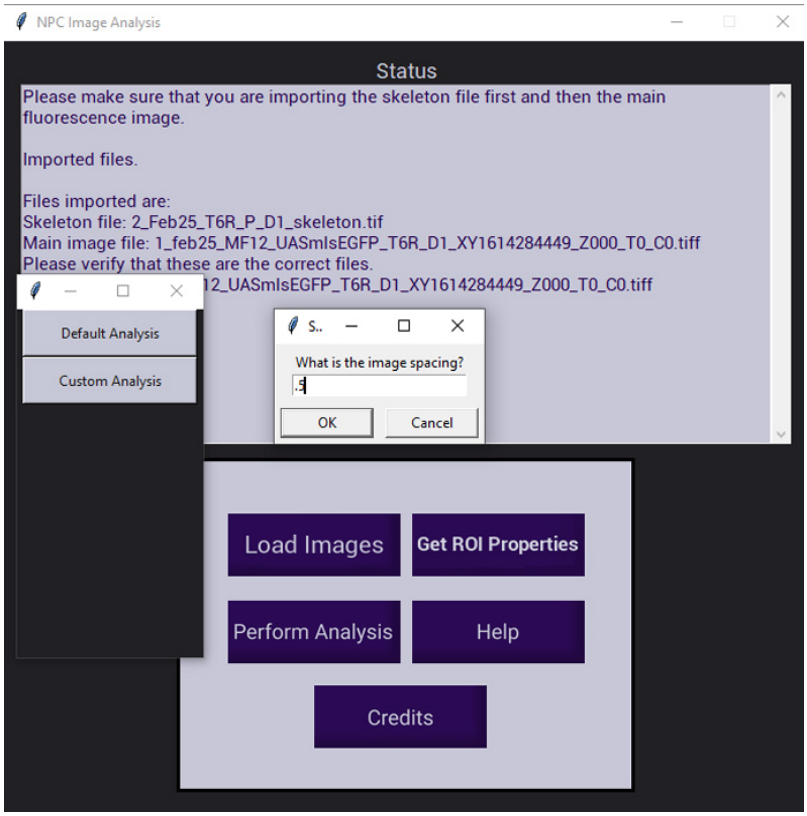
Clicking "Perform Analysis" will open another small window. There are two options to choose for analysis: The "Default Analysis" option and the "Custom Analysis" option.

Default Analysis

When you select "Default Analysis" this will ask you to input the image spacing. This is asking for the "z" spacing of your images.

Note: The spacing indicates the distance between each "z" plane image to create the depth of the image.

For this example, we are using .5:



Pressing "Okay" will initiate the analysis. A napari viewer will open. While this process is occuring, the screen may remain white and the app may be unresponsive as it operates. Don't panic, this is a sign that it is working.

If it remains white and the app is unresponsive for an extended period of time, you should consider the size of your images as this will contribute to the amount of time processing will take.

When the process is complete, you will be able to interact with the napari viewer and see what the final results are.

Note: If you are unhappy with this final result, you can begin a new process with the Custom Analysis option. This will start a new napari viewer and cannot continue with the default analysis as you will be restarting from the beginning.

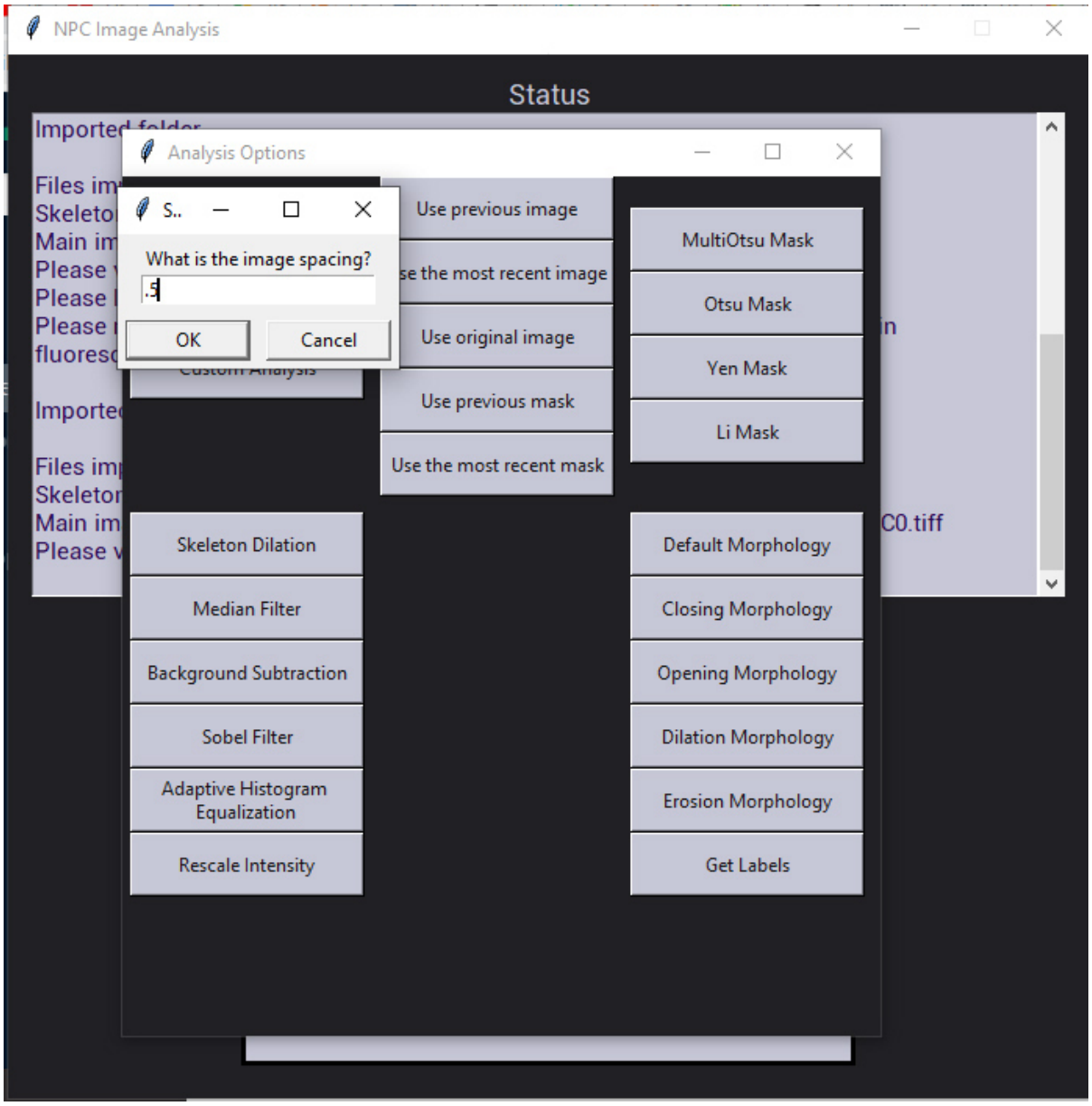
You can follow through the default analysis process and tweak it by visiting the *Default Analysis* page.

Custom Analysis

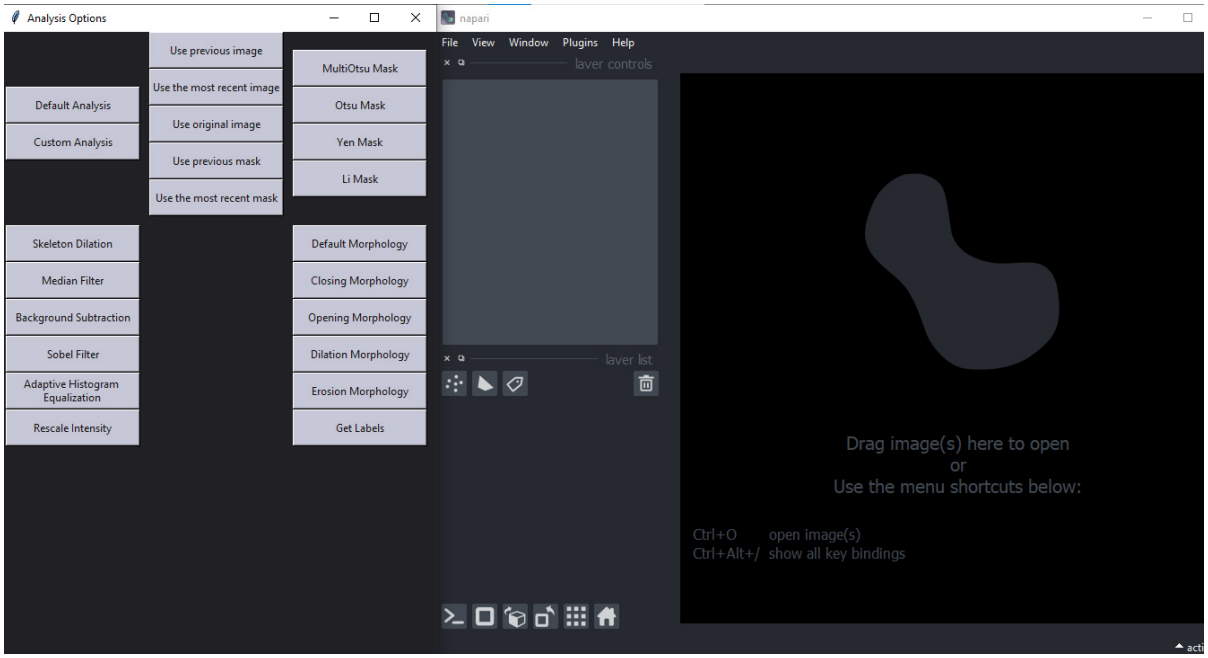
Selecting "Custom Analysis" will expand the current window with the image processing options. Additionally, it will ask the same question as in the default analysis "What is the image spacing?" Again, you will provide the "z" spacing of your images.

Note: The spacing indicates the distance between each "z" plane image to create the depth of the image.

Here we will be using .5:



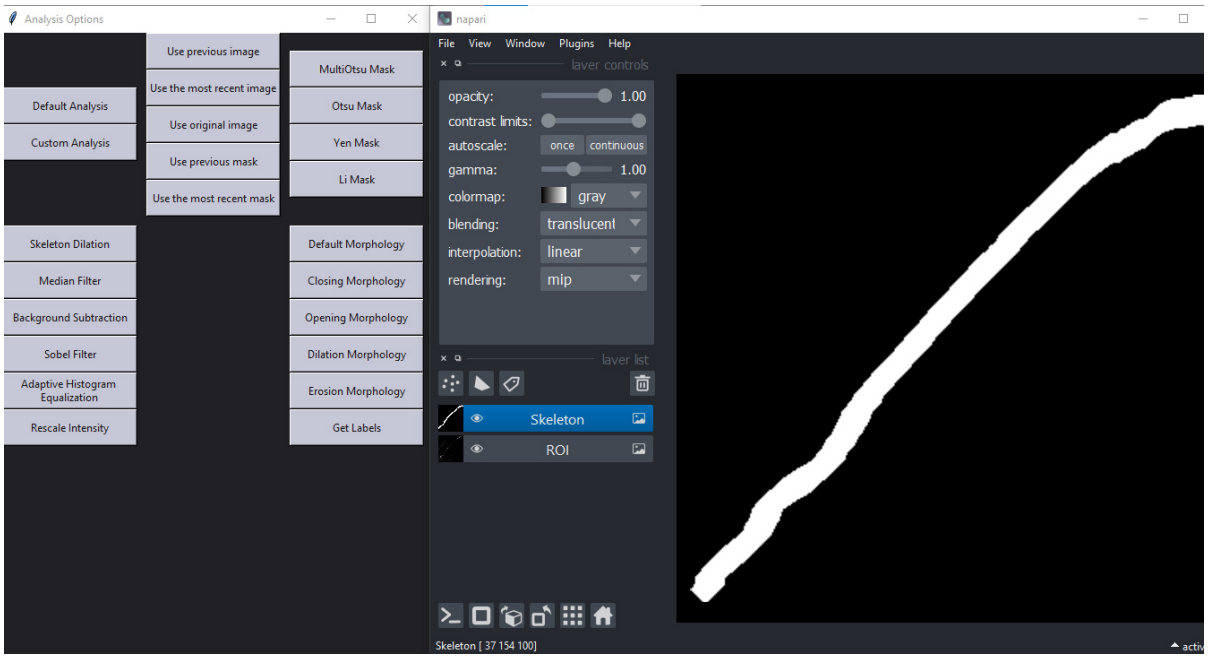
Upon entering your spacing, a napari viewer window will open. It will be empty and allow you the option to begin your image processing.



We will start with a skeleton dilation. Clicking "Skeleton Dilation" will prompt you to input how much you would like to dilate the skeleton image.

The default is 10. For this example, we will also use 10.

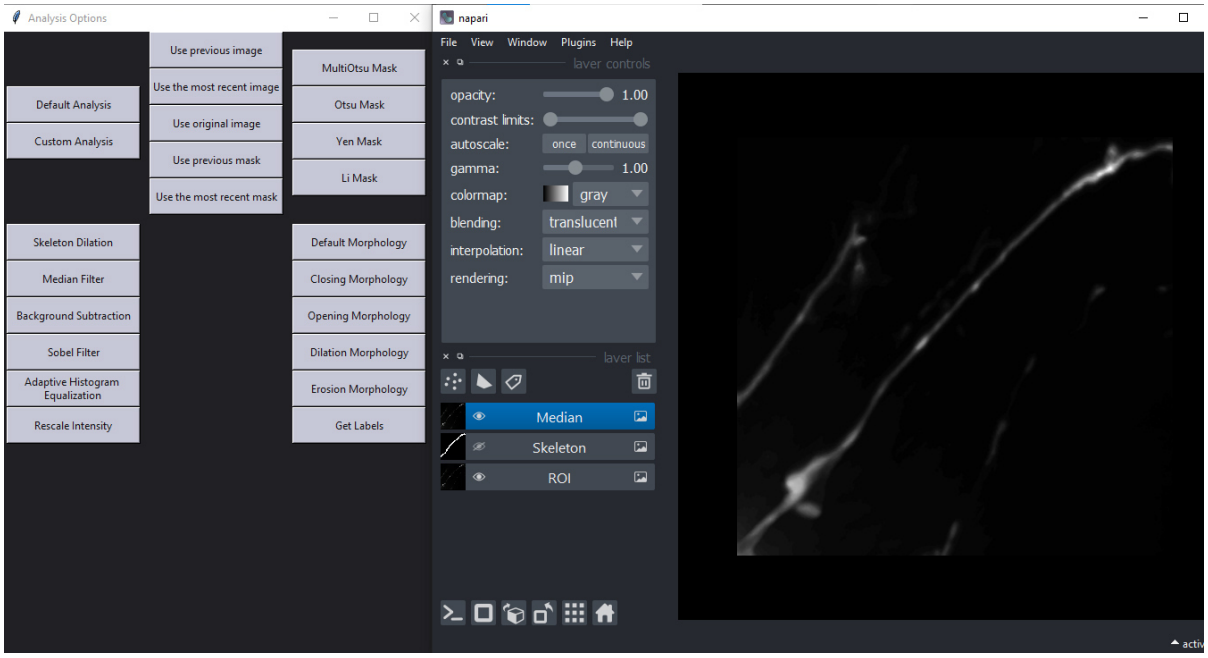
Warning: This operation can take some time, so if it appears that application is frozen, keep in mind the size of the images as this is a contributing factor to the processing power and time consumption of each operation.



This process will return the dilated skeleton as well as the cropped image called 'ROI'. The "status" window will also update with the process that was just performed and what value was performed to allow the user to keep track of what operations they have performed so far.

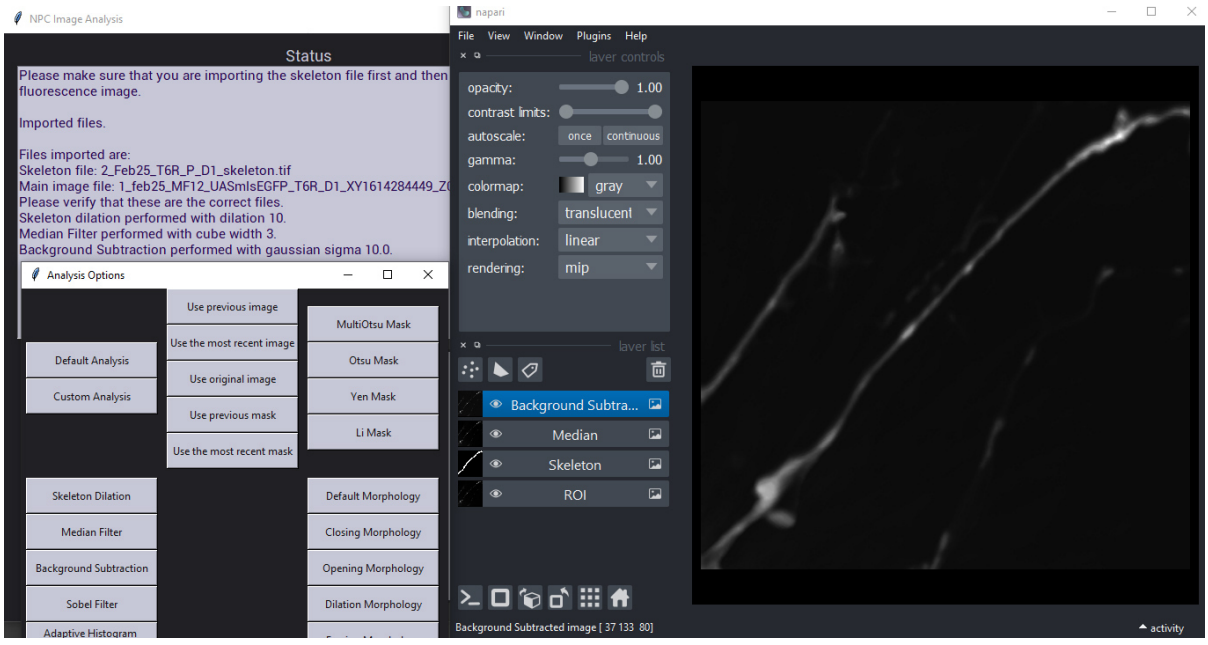
Next, we will apply a median filter. Upon clicking "Median Filter" the application will prompt you to input the cube width to use. The default is 3 which we will use for this example.

This will return the image after the median filter has been applied.



We will apply a background subtraction after this to even out the surrounding background and make it easier for the final step of identify our regions of interest easier. Clicking this will prompt the user to input a gaussian sigma or how strong the blur will be.

The default is 7. In this case, we'll use 10:



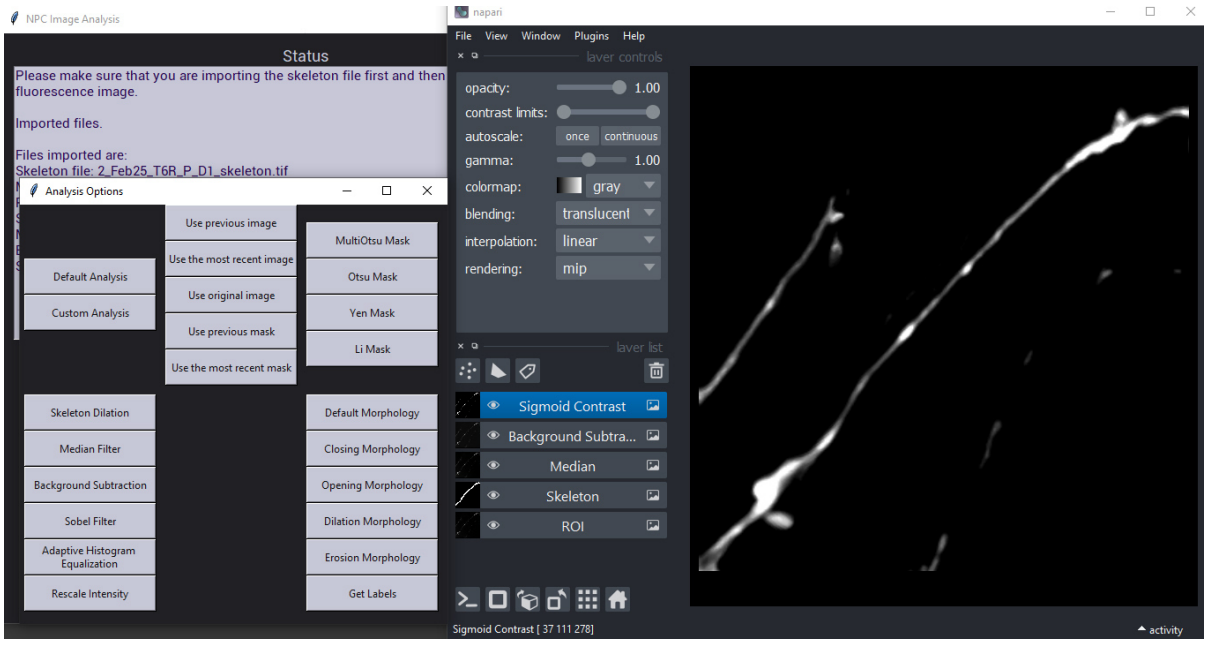
While we could apply the sobel filter which will define the edges of the objects within this image, it's not really necessary here. We want to make the images within the radial process is clear so defining the edges of this particular image may not work in our favor.

We'll enhance the contrast of the image now. One of the byproducts of a background subtraction is that it can also lower the contrast of the overall image as we try to smooth out the background. We'll try the "Rescale Intensity" option first.

Clicking this button will prompt you to enter two values, the minimum intensity value of the image and the maximum intensity values in the image. They are percentages in this case, but the defaults for rescaled intensity is .5 and 99.5 for minimum and maximum respectively.

Note: Choosing the minimum and maximum intensity values means that you will be deciding what the minimum intensity values allowed within the total image allowed and the maximum intensity values allowed in the overall image. If you choose .5 and 99.5, you will be clipping the darkest and brightest 0.5% of pixels within the image thus increasing the overall contrast of the image.

We will use 99.65 and 99.98 for the minimum and maximum values respectively.

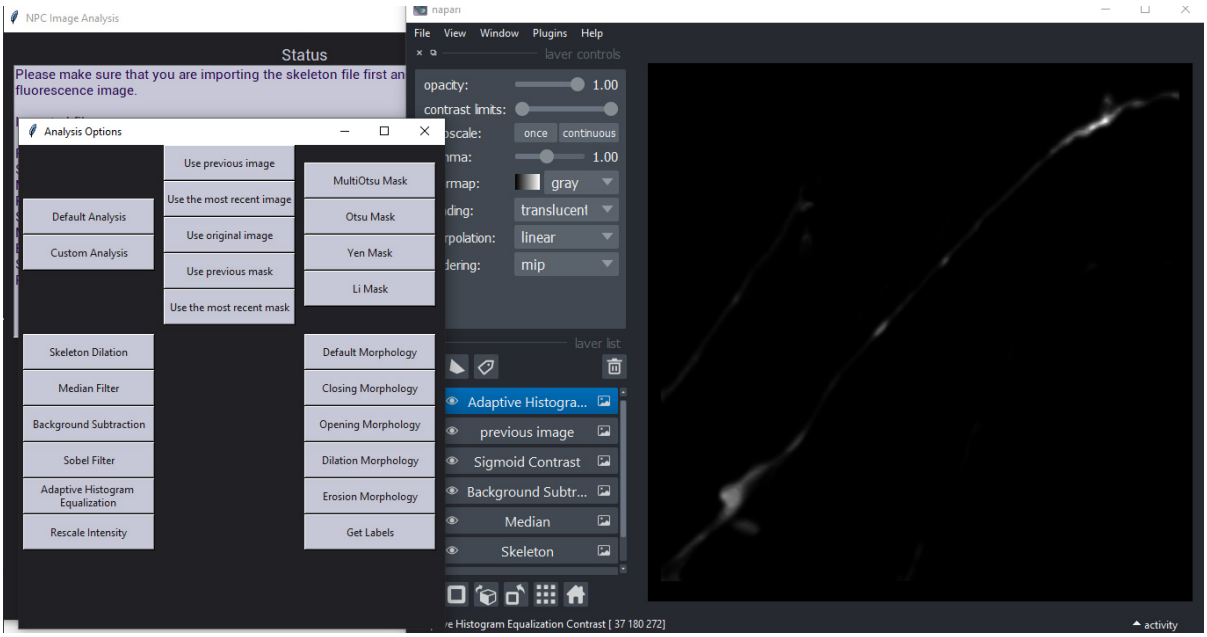


This returns the above. While it is not terrible, it's not exactly what we're looking for right now and adjusting the contrast this way will require a little more playing around.

We'll click the previous image button to go back one step. This will add the previous image we were working with to the viewer once more and the status will update with our selection.

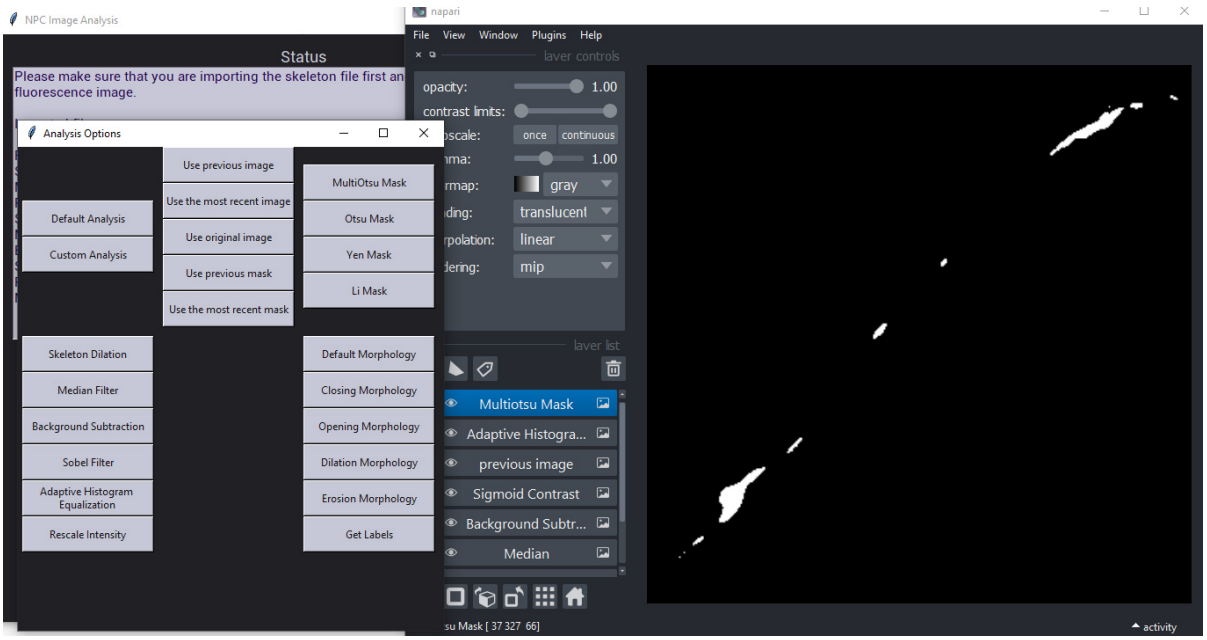
Note: You can only go back one step at the moment.

We should return to the background subtracted image now. We'll go ahead and use the "Adaptive Histogram Equalizaiton" option which will operate automatically. The purpose of this option is to provide an automatic contrast that doesn't require any input from the user. The Rescale Intensity option allows more freedom for the user and can be applied after the adaptive histogram equalization operation is performed to continue to tweak the contrast.



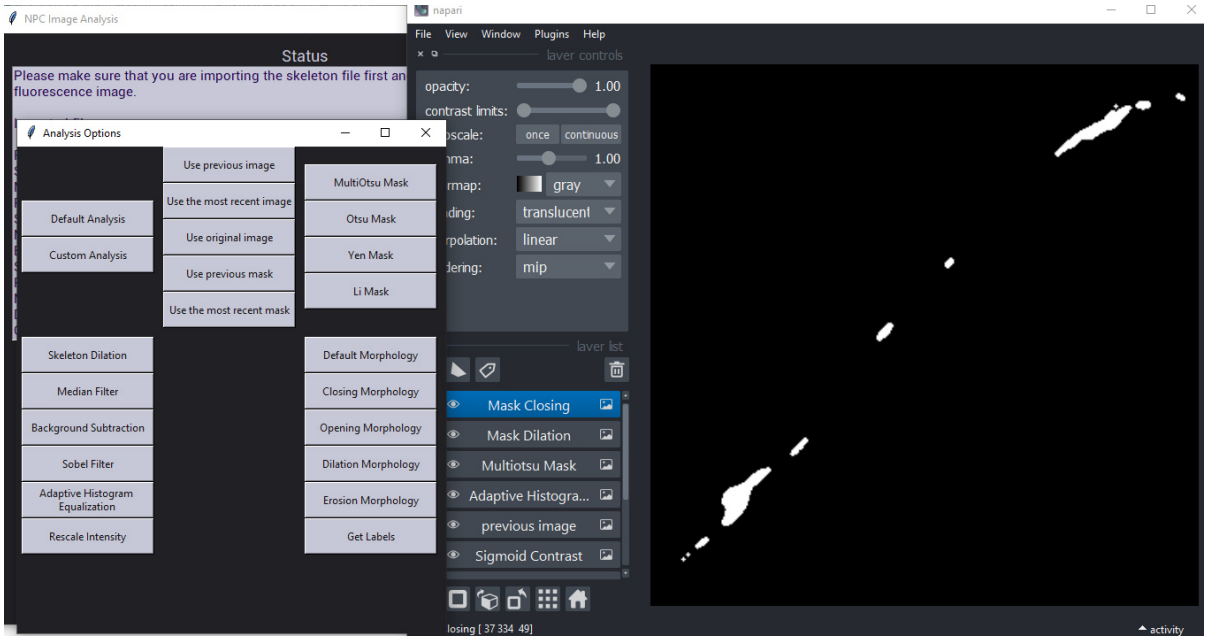
We now have our final contrasted image. From here, it's time to select a mask option. In the default analysis, the multiotsu mask is used. If you don't like the mask that is used you can use the "Use Most Recent Image" option to return to the image prior to using the mask and play around with the options you would like to use.

We'll use the multiotsu mask with 2 classes.

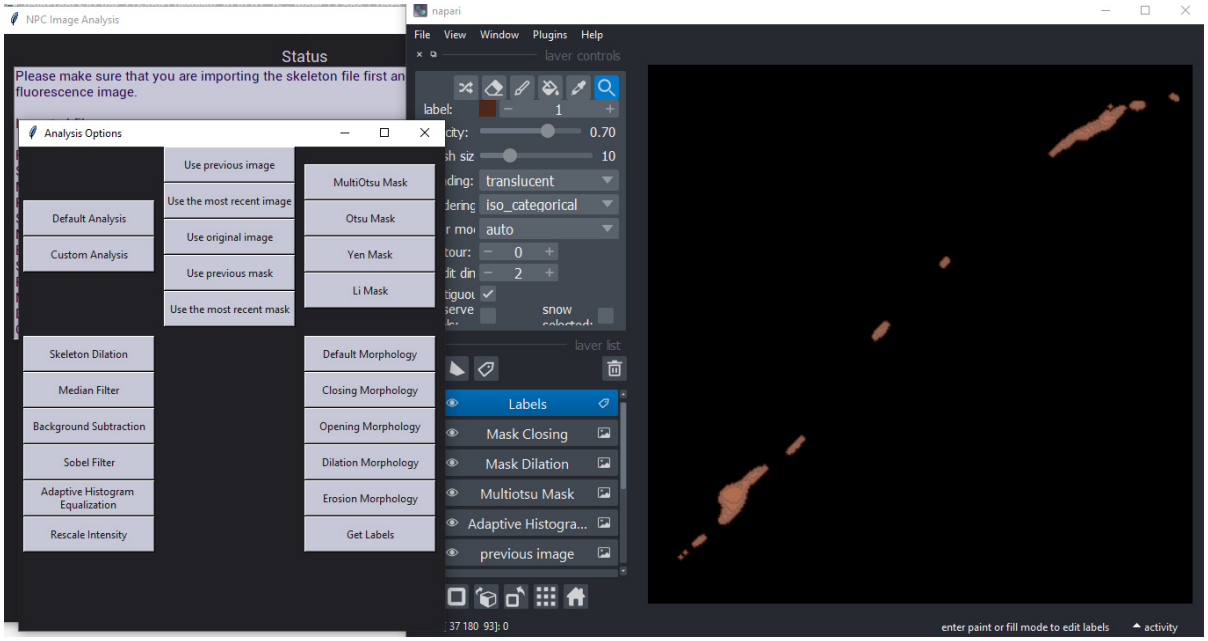


Great! This is looking pretty good, but we can probably define this mask a little bit. The default morphology option allows the user to use the morphological steps that are used in the Default Analysis. The default analysis uses the Dilation and Closing morphological adjustments in that order.

We can see what each of those look like here by using them in that order instead of pressing Default Morphology.



This final mask looks pretty good. Once you're pleased with the final mask, you can generate labels by pressing the "Get Labels" options. Labels will be generated and added to the viewer.



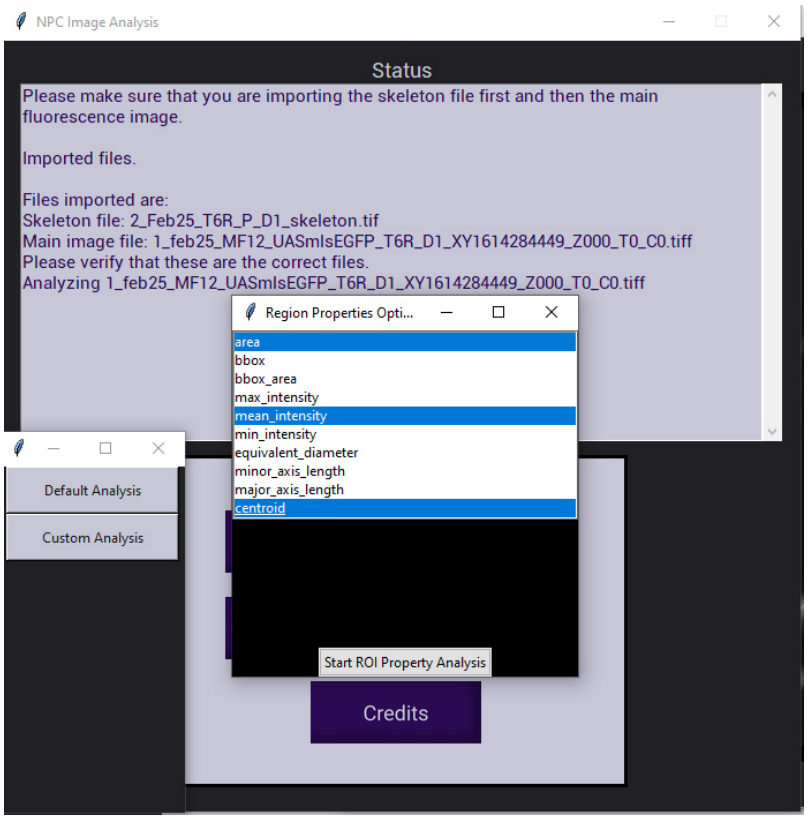
From here, we can start getting data from our analysis.

2.5.3 Getting ROI Properties

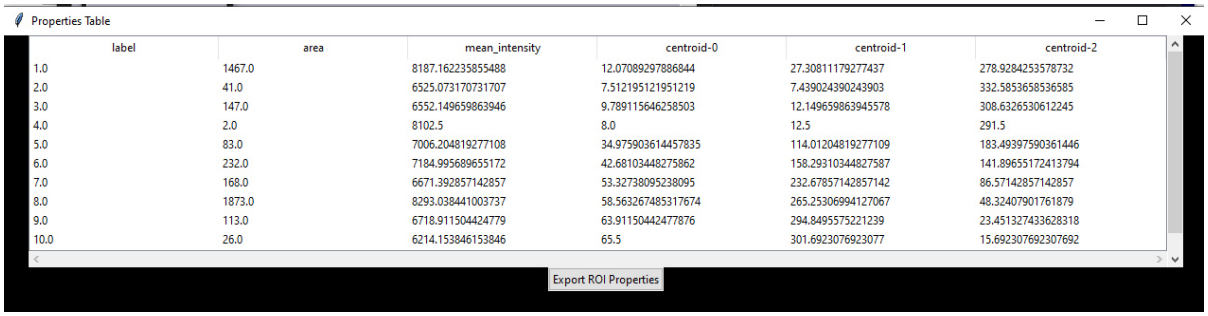
From here you can begin the process of obtaining properties of the identified regions of interest (ROI) by clicking the "Get ROI Properties" button.

Clicking this button will open up a new window allowing you to select the data you would like to obtain. For this example, we'll select "area", "mean_intensity", and "centroids".

After that, click the "Start ROI Properties Analysis" button.



After this process is completed, you will be presented with a table containing data for each region of interest and the properties you selected.



Clicking the "Export ROI Properties" button will open the save dialog and allow you to save your data. If you would like to get additional properties, you can return to the window with your property options and generate a new table.

2.5.4 Example Data Returned

Note: This does not include all of the data that can be returned. Additionally, all data returned regarding points or coordinates (this includes bbox, centroids, coords) are returned in z, x, y order (plane, row, column)

Table 1: Sample Data

label	area	mean_intensity	centroid-0	centroid-1	centroid-2
1	1467	8187.162236	12.07089298	27.30811179	278.9284254
2	41	6525.073171	7.512195122	7.43902439	332.5853659
3	147	6552.14966	9.789115646	12.14965986	308.6326531
4	2	8102.5	8	12.5	291.5
5	83	7006.204819	34.97590361	114.0120482	183.4939759
6	232	7184.99569	42.68103448	158.2931034	141.8965517
7	168	6671.392857	53.32738095	232.6785714	86.57142857
8	1873	8293.038441	58.56326749	265.2530699	48.32407902
9	113	6718.911504	63.91150442	294.8495575	23.45132743
10	26	6214.153846	65.5	301.6923077	15.69230769
11	12	6545.666667	66.5	305	12

The data above is generated using sci-kit image's regionprops table module. If you would like to read more about it and the function you can find it here.

Image_Analysis_App

Image_Analysis

2.6 Image_Analysis_App

Classes

ImageSeg_App(master)	A class used to create the Image Segmentation App

2.6.1 Image_Analysis_App.ImageSeg_App

class Image_Analysis_App.**ImageSeg_App**(*master*)

Bases: `object`

A class used to create the Image Segmentation App

…

master [TK()] TK toplevel object to initialize the application

credits() Prints the credits for this application on the status screen

help_info() Prints help information for this application based on the user's choice

read_files() Allows the user to select the images needed for analysis

bttn(window, text, bcolor, fcolor, command) window : top level object text : str bcolor : str fcolor : str command : method Creates the buttons for the custom analysis window

default_analysis() Performs the default analysis operations for the images provided.

custom_analysis() Allows the user to customize the analysis for their images

analyze() Creates the window for the user to choose their analysis option.

recent_image() Allows the user to access the most recently created image.

previous_image() Allows the user to access the image created prior to the most recent one

original_image() Allows the user to access the original image essentially resetting their analysis.

recent_mask() Allows the user to access the most recently created mask

previous_mask() Allows the user to access the mask created prior to the most recent mask

skeleton_dilate() Allows the user to dilate the skeleton mask loaded in to isolate their region of interest

median_image() Allows the user to perform a median filter operation through the customization window

background_subtract() Allows the user to perform a background subtract operation through the customization window

sobel_image() Allows the user to perform a sobel filter operation through the customization window

ahe() Allows the user to perform an adaptive histogram equalization operation through the customization window

rescale_intensity() Allows the user to perform a rescaled intensity operation through the customization window

multiotsu() Allows the user to perform a multiotsue mask operation through the customization window

otsu() Allows the user to perform an otsu mask operation through the customization window

yen() Allows the user to perform a yen mask operation through the customization window

li() Allows the user to perform a li mask operation through the customization window

defaul_morph() Allows the user to use the default morphology operations used in the default analysis

closing_morph() Allows the user to use the closing morphology operation through the customization window

opening_morph() Allows the user to use the opening morphology operation through the customization window

dilation_morph() Allows the user to use the dilation morphology operation through the customization window

erosion_morph() Allows the user to use the erosion morphology operation through the customization window

properties_table() Creates the table with properties from the analysis as selected by the user

props() Allows the user to select what properties they would like to include in their analysis

export() Exports the properties table as an excel file

`__init__`(*master*)

Methods

__init__(master)	

ahe()	Performs the adaptive histogram equalization operation from scikit image.
analyze()	Intialize the window for the user to choose their analysis option: Default or Custom
background_subtract()	Function for the background subtraction button.
bttn(window, text, bcolor, fcolor, command)	Creates the buttons for the custom analysis.
closing_morph()	Perfroms the closing morphology operation.
credits()	Returns the credits for this program.
custom_analysis()	Function for the user to customize their analysis process.
default_analysis()	Default analysis operation for loaded image.
default_morph()	Performs the default morphology operation.
dilation_morph()	Performs the dilation morphology operation.
erosion_morph()	Performs the erosion morphology operation.
export()	Exports an excel sheet of the region propery information obtained.
help_info()	Provides some help information about how to use the program.
label_image()	This will create labels for the image.
li()	Li mask morphology operation.
median_image()	Function for the median filter button.
multiotsu()	MultiOtsu mask morphology operation.
opening_morph()	Performs the opening morphology operation.
original_image()	Accesses the original image, esentially resetting the analysis
otsu()	Otsu mask morphology operation.
previous_image()	Accesses the image created prior to the most recent one
previous_mask()	Accesses the mask created prior to the most recent one
properties_table()	Creates the table of properties for analysis.
props()	Obtains properties of the identified regions of interest.
read_files()	Loads the images in to perform analysis on.
recent_image()	Accesses the most recent image created.
recent_mask()	Accesses the most recent mask created.
rescale_intensity()	Performs the rescale intensity operation from scikit image.
skeleton_dilate()	Function to dilate the skeleton loaded in by the user to isolate to their region of interest.
sobel_image()	Performs the sobel operation from scikit image.
yen()	Yen mask morphology operation.

ahe()

Performs the adaptive histogram equalization operation from scikit image.

In addition, this function will multiply the adaptive histogram equalization image will be multiplied to the previous image passed in to enhance the contrast.

analyze()

Intialize the window for the user to choose their analysis option: Default or Custom

background_subtract()

Function for the background subtraction button. Subtracts the background from the image.

Asks the user for the gaussian sigma value to use. The default value is 7. If the user does not provide a value, will simply cancel the operation.

bttn(*window*, *text*, *bcolor*, *fcolor*, *command*)

Creates the buttons for the custom analysis.

closing_morph()

Perfroms the closing morphology operation.

credits()

Returns the credits for this program.

custom_analysis()

Function for the user to customize their analysis process.

This contains all of the possible analysis options available to the user. This initializes a different napari viewer from the default analysis, and will stop the default analysis viewer from continued use if one was created prior to this.

default_analysis()

Default analysis operation for loaded image.

Will initialize the image analysis function with the default parameters without any customization. Creates a napari viewer used only for the default analysis.

default_morph()

Performs the default morphology operation.

This operation consists of dilation and closing morphology steps.

dilation_morph()

Performs the dilation morphology operation.

erosion_morph()

Performs the erosion morphology operation.

export()

Exports an excel sheet of the region propery information obtained. This exports the table created from the proerties_table function

help_info()

Provides some help information about how to use the program.

label_image()

This will create labels for the image.

The returned labels are regions of interests determined by the final mask after all of the analysis is done. Getting these labels will set analysis to complete and allow for properties to be exported

li()

Li mask morphology operation.

Uses the current image and doesn't require any args.

median_image()

Function for the median filter button. Performs the median filter operation from scikit image.

Asks the user for the cube width value to use. The default value is 3. If the user does not provide a value, will simply cancel the operation.

multiotsu()

MultiOtsu mask morphology operation.

Uses the current image and doesn't require any args.

opening_morph()

Performs the opening morphology operation.

original_image()

Accesses the original image, esentially resetting the analysis

otsu()

Otsu mask morphology operation.

Uses the current image and doesn't require any args.

previous_image()

Accesses the image created prior to the most recent one

previous_mask()

Accesses the mask created prior to the most recent one

properties_table()

Creates the table of properties for analysis.

This is created from the labels generated and based on the properties you selected.

props()

Obtains properties of the identified regions of interest.

This is based on the labels obtained. This step cannot be performed if the analysis is set to complete. Properties can be changed based on what the user selects when prompted

read_files()

Loads the images in to perform analysis on.

recent_image()

Accesses the most recent image created.

recent_mask()

Accesses the most recent mask created.

rescale_intensity()

Performs the rescale intensity operation from scikit image.

Asks the user for the minimum and maximum intensity value to use. The default values are 0 (minimum) and 255 (maximum). If the user does not provide a value, will simply cancel the operation.

skeleton_dilate()

Function to dilate the skeleton loaded in by the user to isolate to their region of interest.

sobel_image()

Performs the sobel operation from scikit image.

yen()

Yen mask morphology operation.

Uses the current image and doesn't require any args.

2.7 Image_Analysis

Classes

Image_Processing(image, skeleton, spacing, ...)	A class that will perform image processing operations.

2.7.1 Image_Analysis.Image_Processing

class Image_Analysis.**Image_Processing**(*image*, *skeleton*, *spacing*, *viewer*)

Bases: object

A class that will perform image processing operations.

...

image: ndarray an image to be processed

skeleton: ndarray the skeleton of the image to be processed

spacing: float the z spacing of the image

viewer: napari.viewer the viewer to be used to view the image layers in 3D

skeleton_dilation(dilation=10) Dilates the skeleton of the image as many times as you indicate. Crops the image to only the region of interest

get_cropped() returns the cropped image

median_filter(image, cube_width) Performs the median filter operation from skimage on the image passed. Uses the cube width provided.

background_subtract(image, gauss_sigma) Subtracts the background from the passed in image. Uses the gaussian filter from skimage on the image passed in.

sobel(image) Performs the sobel transformation on the passed in image using the sobel filter from skimage

ahe_contrast(image) Performs the adaptive histogram equalization on the passed in image using the equalize_adapthist from skimage. Multiplies the image to the image passed in to improve the contrast

rescaled_intensity(image, vmin, vmax) Performs the rescale intensity operation on the passed in image using the operation from skimage

multiotsu_mask(image, classes) Performs the multiotsu mask operation using threshold_multiots from skimage using the number of classes specified

otsu_mask(image) Performs the otsu mask operation using threshold_otsu from skimage

li_mask(image) Performs the li mask operation using threshold_li from skimage

yen_mask(image) Performs the yen mask operation using threshold_yen from skimage

default_morphology(image) Uses a default morphology operation consisting of closing and dilation morphological operations.

closing(image) Performs the closing operation on the provided mask

opening(image) Performs the opening operation on the provided mask

dilation(image) Performs the dilation operation on the provided mask

erosion(image) Performs the erosion operation on the provided mask

labels(image) Generates labels based on the mask created

region_props(labels, intensity_image, properties) Generates the region properties from the labels depending on the properties selected

`__init__`(*image*, *skeleton*, *spacing*, *viewer*)

image [ndarray] The image to use for analysis

skeleton [ndarray] The binary image used to identify the area of interest of the image

spacing [float] The z spacing of the images being passed

viewer [napari viewer] The viewer to use for visualization of the images

Methods

__init__(image, skeleton, spacing, viewer)	Parameters image ndarray The image to use for analysis skeleton ndarray The binary image used to identify the area of interest of the image spacing float The z spacing of the images being passed viewer napari viewer The viewer to use for visualization of the images
ahe_contrast([image])	Performs the adaptive histogram equalization operation and multiplies the provided image and the AHE image
background_subtract([image, gauss_sigma])	Subtracts the background from the image provided
closing([mask])	Performs the closing morphology operation from skimage
default_morphology()	Performs the default morphology operations: Binary Dilation and Binary Closing from skimage
dilation([mask])	Performs the dilation morphology operation from skimage
erosion([mask])	Performs the erosion morphology operation from skimage
get_cropped()	Returns the cropped image used in the analysis.
labels([mask])	Generates labels based on the provided mask
li_mask([image])	Performs the li mask operation from skimage
median_filter([image, cube_width])	Performs the median filter operation from skimage on the image passed.
multiotsu_mask([image, classes])	Performs the multiotsu mask operation from skimage
opening([mask])	Performs the opening morphology operation from skimage
otsu_mask([image])	Performs the otsu mask operation from skimage
region_props(labels, intensity_image, properties)	Generates a dictionary of properties requested for the labels provided
rescaled_intensity([image, vmin, vmax])	Rescales the intensity of the image using the intensities provided
skeleton_dilation([dilation])	Dilates the skeleton provide.
sobel([image, mask])	Performs the sobel transformation on a given image
yen_mask([image])	Performs the yen mask operation

ahe_contrast(*image=None*)

Performs the adaptive histogram equalization operation and multiplies the provided image and the AHE image

image [ndarray] The image to apply the operation to

final_contrast An ndarray of of the same size as the input after the AHE and multiplication operations have been applied

`background_subtract`(*image=None*, *gauss_sigma=7*)

Subtracts the background from the image provided

If the argument *gauss_sigma* isn't provided, the default value of 7 is passed.

image [ndarray] The image to apply the operation to

gauss_sigma [int, optional] The sigma used for the gaussian operation (default is 7)

subtract The image with the same shape as the input image after the subtraction has been applied

`closing`(*mask=None*)

Performs the closing morphology operation from skimage

mask [ndarray] The mask to perform the morph operation on

mask_morph The final mask of type ndarray after the morphology operation is performed

`default_morphology`()

Performs the default morphology operations: Binary Dilation and Binary Closing from skimage

mask_closing an array of the same shape as the input image. The final mask after the morphology operations are performed

final_mask an array of the same shape as the input image. The final mask isolated only to the area within the skeleton

`dilation`(*mask=None*)

Performs the dilation morphology operation from skimage

mask [ndarray] The mask to perform the morph operation on

mask_morph The final mask of type ndarray after the morphology operation is performed

`erosion`(*mask=None*)

Performs the erosion morphology operation from skimage

mask [ndarray] The mask to perform the morph operation on

mask_morph The final mask of type ndarray after the morphology operation is performed

`get_cropped`()

Returns the cropped image used in the analysis.

`labels`(*mask=None*)

Generates labels based on the provided mask

mask [ndarray] The mask to perform the morph operation on

boundaries an array of the same shape as the input

label_image an array with labels of the same shape as the input. All connected areas are under one label

`li_mask`(*image=None*)

Performs the li mask operation from skimage

image [ndarray] The image to apply the operation to

mask_morph an array with the same shaqpe as the input after the mask has been applied

`median_filter`(*image=None*, *cube_width=3*)

Performs the median filter operation from skimage on the image passed.

If the argument *cube_width* isn't provided, the default value of 3 is passed.

image [ndarray] The image to apply the operation to

cube_width [int, optional] The width of the cube used as the footprint for the filter (default is 3)

median The image after the median filter has been applied

`multiotsu_mask`(*image=None*, *classes=2*)

Performs the multiotsu mask operation from skimage

If the argument *classes* isn't provided, the default value of 2 is passed.

image [ndarray] The image to apply the operation to

classes [int, optional] The number of classes in the image (default is 2)

mask_morph an array with the same shaqpe as the input after the mask has been applied

`opening`(*mask=None*)

Performs the opening morphology operation from skimage

mask [ndarray] The mask to perform the morph operation on

mask_morph The final mask of type ndarray after the morphology operation is performed

`otsu_mask`(*image=None*)

Performs the otsu mask operation from skimage

image [ndarray] The image to apply the operation to

mask_morph an array with the same shaqpe as the input after the mask has been applied

`region_props`(*labels*, *intensity_image*, *properties*)

Generates a dictionary of properties requested for the labels provided

labels [ndarray] The labeled image to generate the properties for

intensity_image [ndarray] The image to reference intensity values for

properties [list] A list of strings with the properties desired

props a dictionary with the properties generated for each label

`rescaled_intensity`(*image=None*, *vmin=0.5*, *vmax=99.5*)

Rescales the intensity of the image using the intensities provided

If the arguments *vmin* and *vmax* aren't provided, the default values of 0 and 255 are passed.

image [ndarray] The image to apply the operation to

vmin [int, optional] The minimum intensity value allowed for the image (default is 0)

vmax [int, optional] The maximum intensity value allowed for the image (default is 255)

final_contrast An ndarray of of the same size as the input after the rescaled intensity operation has been applied

`skeleton_dilation`(*dilation=10*)

Dilates the skeleton provide.

If the argument *dilation* isn't passed in, the default dilation value is used. Crops the image to be analyzed to keep the image size small.

dilation [int, optional] The amount of dilation applied to the skeleton (default is 10)

ROI An array for the cropped image with the dimensions of the skeleton

cropped_skeleton An array for the cropped skeleton with the size of the skeleton

`sobel`(*image=None*, *mask=None*)

Performs the sobel transformation on a given image

If the argument *mask* isn't provided, the mask will not be used.

image [ndarray] The image to apply the operation to

mask [ndarray, optional] mask to isolate the sobel filter to (default is None)

sobel The image of same shape as the input after the sobel filter has been applied

`yen_mask`(*image=None*)

Performs the yen mask operation

image [ndarray] The image to apply the operation to

mask_morph an array with the same shaqpe as the input after the mask has been applied

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

i

INDEX

Symbols

A

B

C

D

E

G

H

I

L

M

O

P

R

S

Y