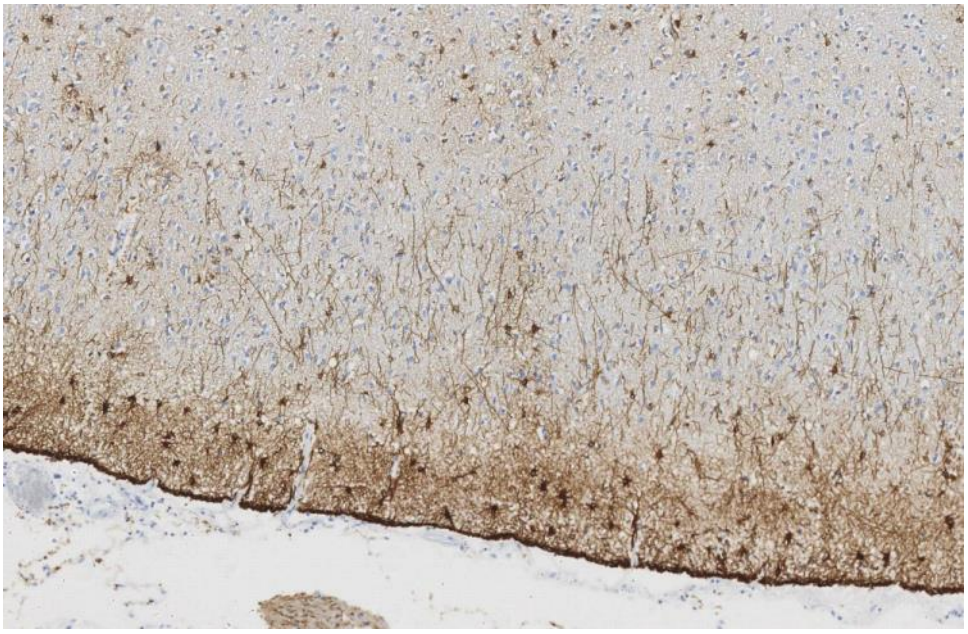
2017-03-22 (1)
Color Deconvolution

Τετάρτη, 22 Μαρτίου 2017 12:06 μμ

COLOR DECONVOLUTION PLUGIN

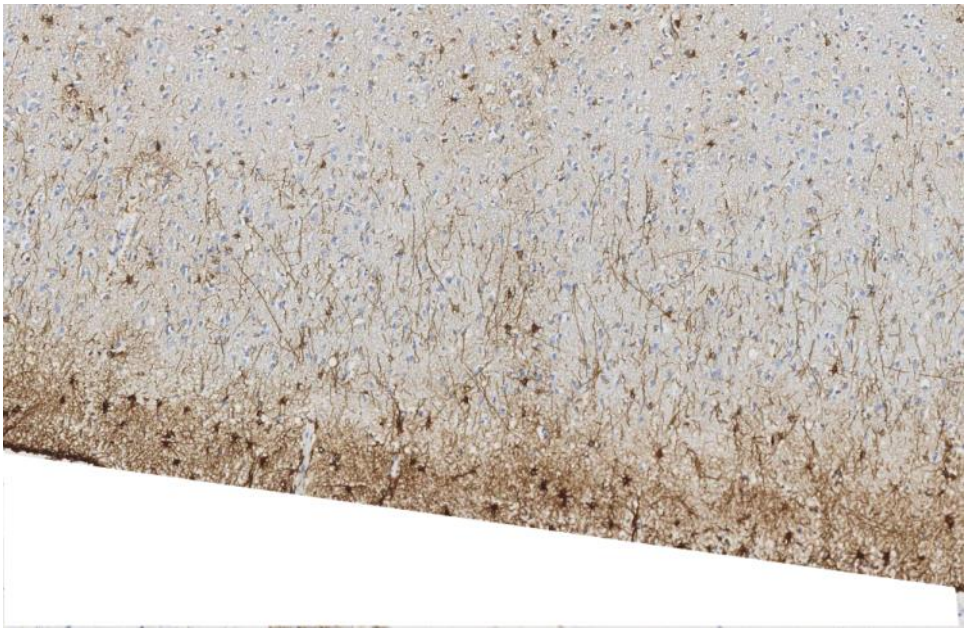
We begin with testing a small part of an original image that we have from our stained samples (test-img-GFAP).



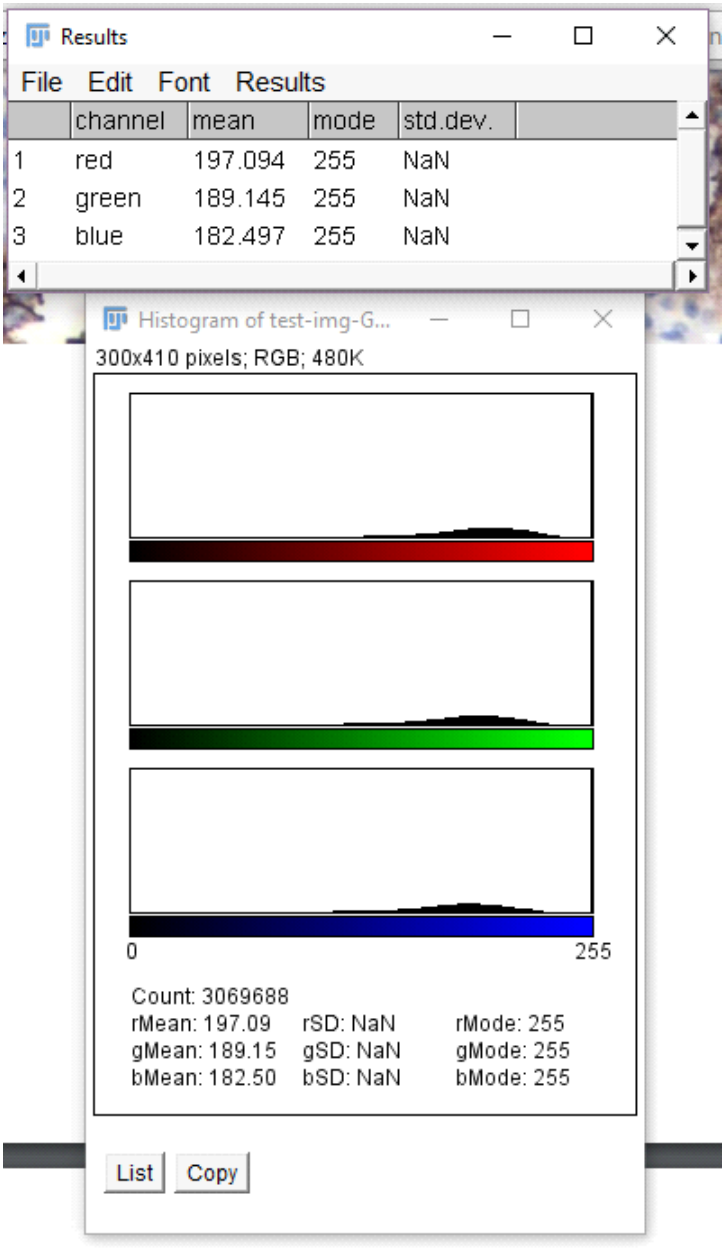
We want to analyze the image by various aspects but the first thing would be to test the color deconvolution analysis.
To achieve that we use the **latest version of FIJI**.

We will try to separate the image in **two bases of color**. An image based on **blue** and an image based on **brown**.

First thing we do is to open the image in FIJI. Now, in order to start processing we have to make sure that the **background is neutral** in order to avoid possible artifacts in our results.
So we create a ROI (Region of Interest) for the part of the image that we don't want to include and by using the basic tools of FIJI we draw it white (**Polygon Selection -> ROI -> (right click) Fill**)
The image we create is the following:

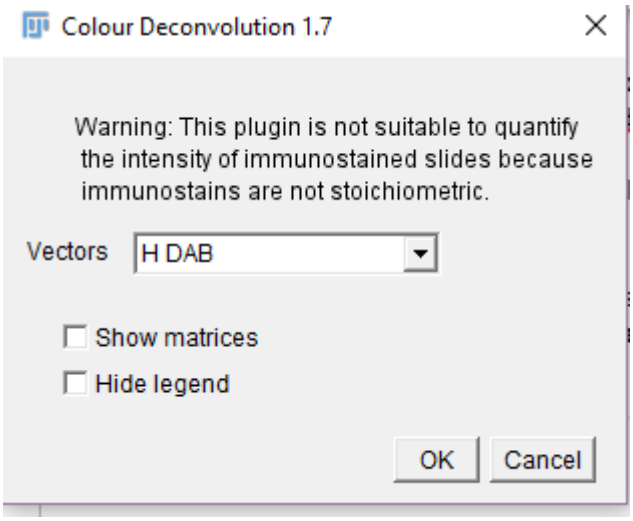


Next thing is to ensure that the RGB values are of similar number. To test that we use (**Analyze -> Color Histogram**). In our case the RGB values are similar for all three colors.

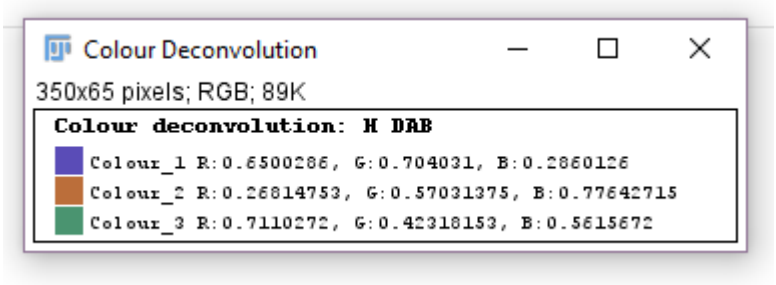


*Though, **in case they were not** we would have to adjust the image properly: **Process -> subtract background** or **Image -> Adjust processes**.

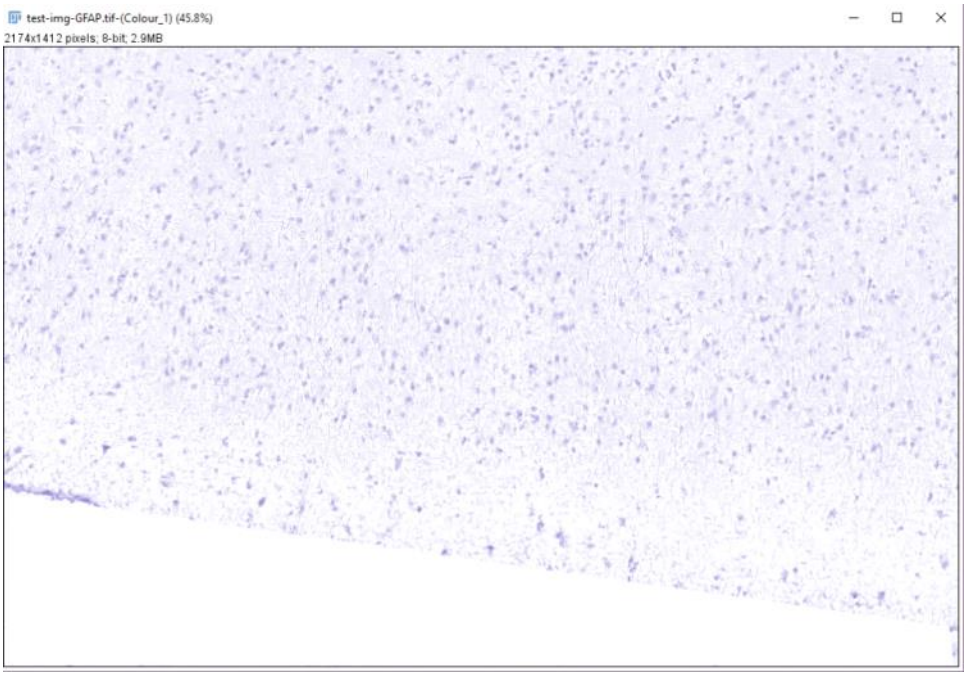
Following, we can use the Color Deconvolution plugin tool in order to separate the image to the desired colors: **Image -> Color -> Color Deconvolution.**
In the window that pops up we have to select the stain method that we are working on from the list given. (for the specific sample) we select the **"H DAB"** option.



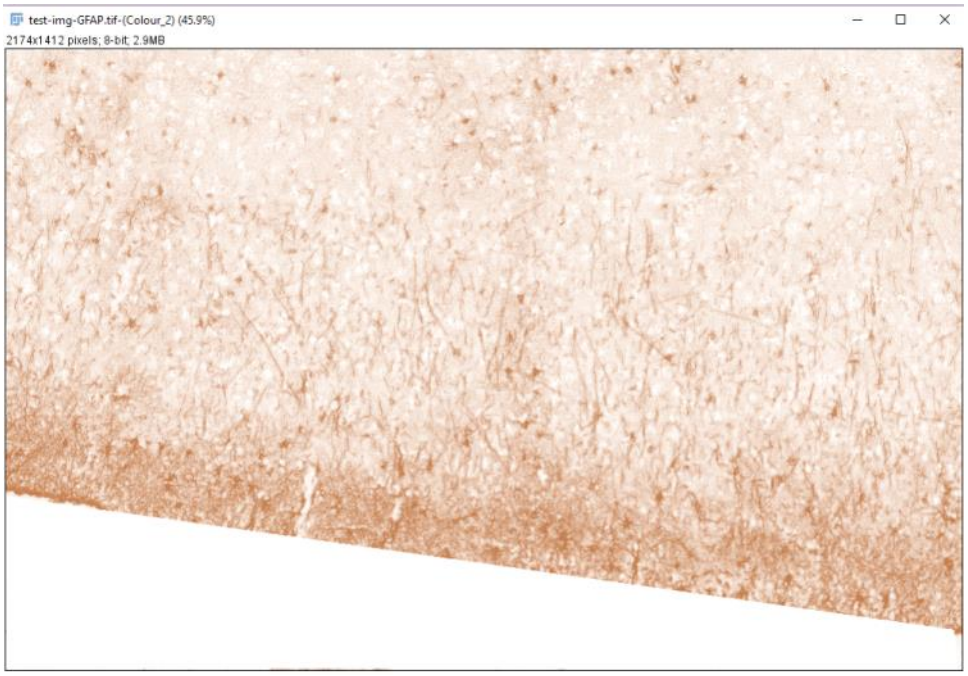
4 new windows show up. Three for the new images created (1 blue, 1 brown and 1 white). The fourth is a window with values of the three new images.



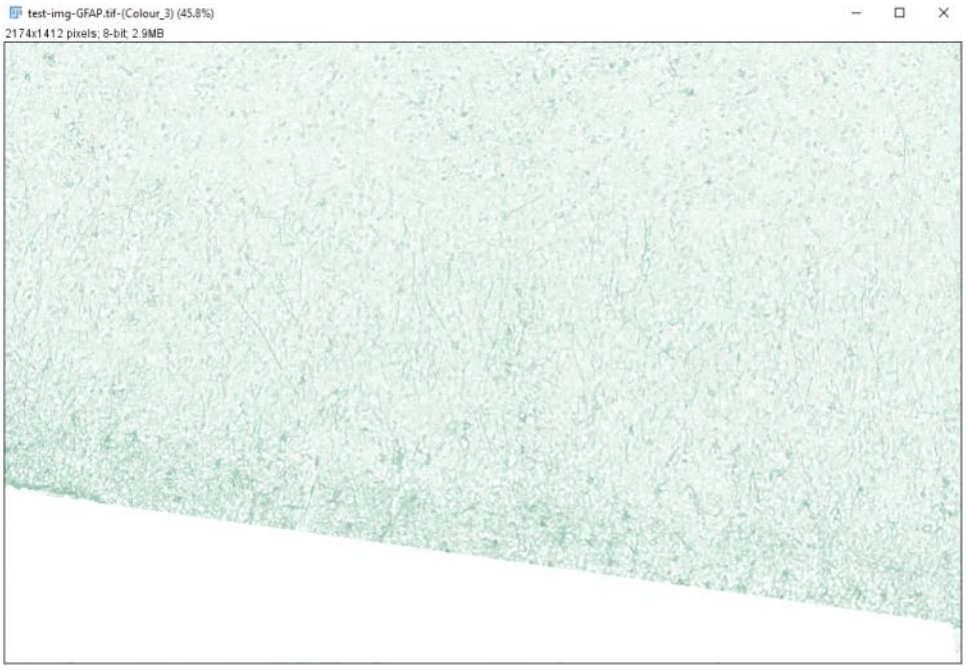
Blue:



Brown:



White:



Now we are in position of comparing the images in order to find out in each anything we are interested in.

*Some useful information about the **"Color Deconvolution plugin"** can be found here:

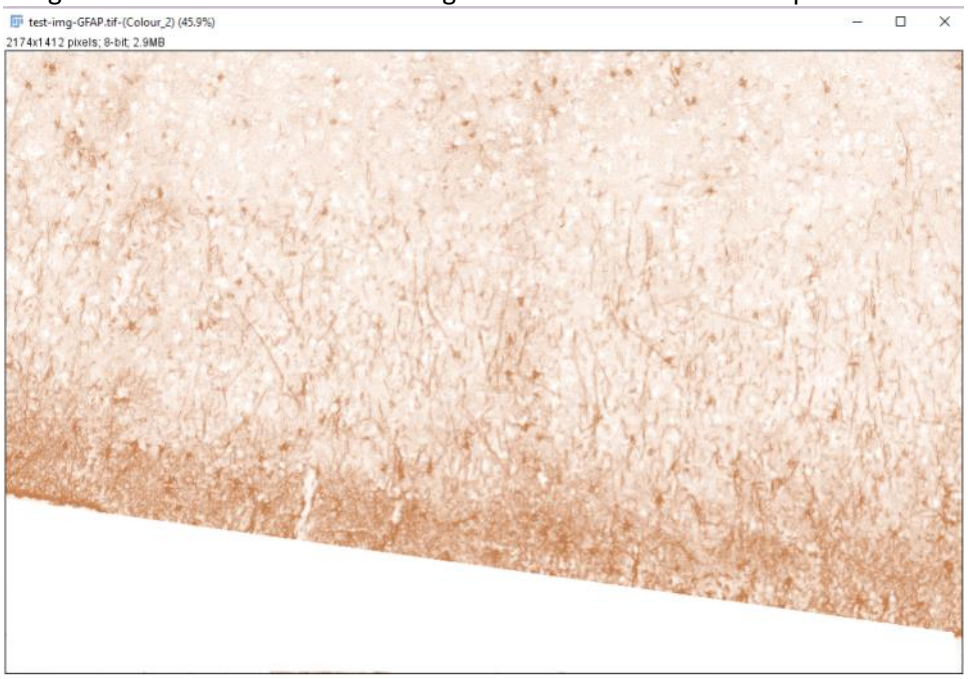
Using Colour De...
https://www.fmhs.auckland.ac.nz/assets/fmhs/sms/biru/docs/Using%20Colour%20Deconvolution%20in%20ImageJ.pdf

2017-03-22 (2)
Tubeness

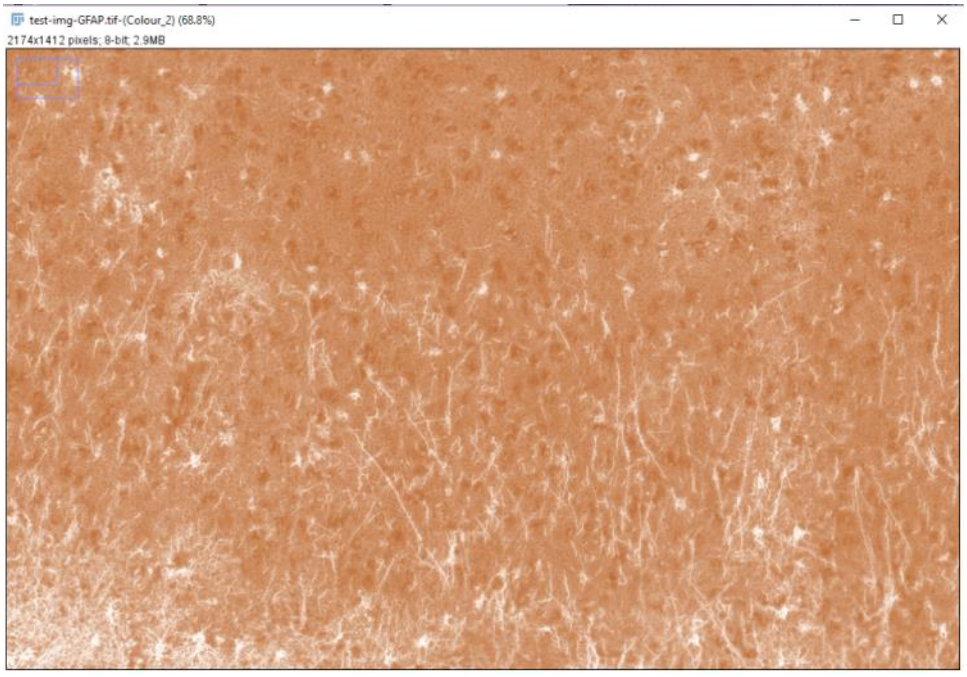
Τετάρτη, 22 Μαρτίου 2017 5:58 μμ

TUBENESS PLUGIN

This is useful as a preprocessing step for tracing neurons. In our case we will use it on our "brown" image. So first of all we have the image that we are interested in open:

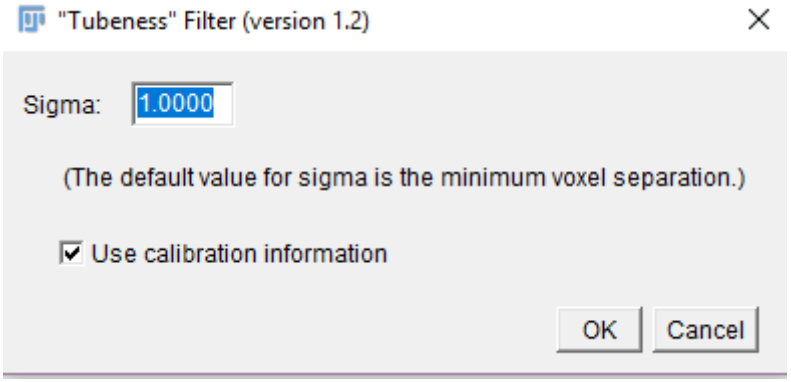


IMPORTANT NOTE: Once we need to distinguish the "tubes" we have to be aware that the regions of interest within the image HAVE TO BE bright (because this is what the plugin is looking to measure). If we run the "**look up table (LUT)**" we can see that the bright values are in fact the background of what we want to measure and not the neurons of which the structure we want to measure (which are the dark values). What we have to do is "**invert**" (inverts the bright and dark values of the image) **"Edit -> Invert**". So our new image of interest is the following:

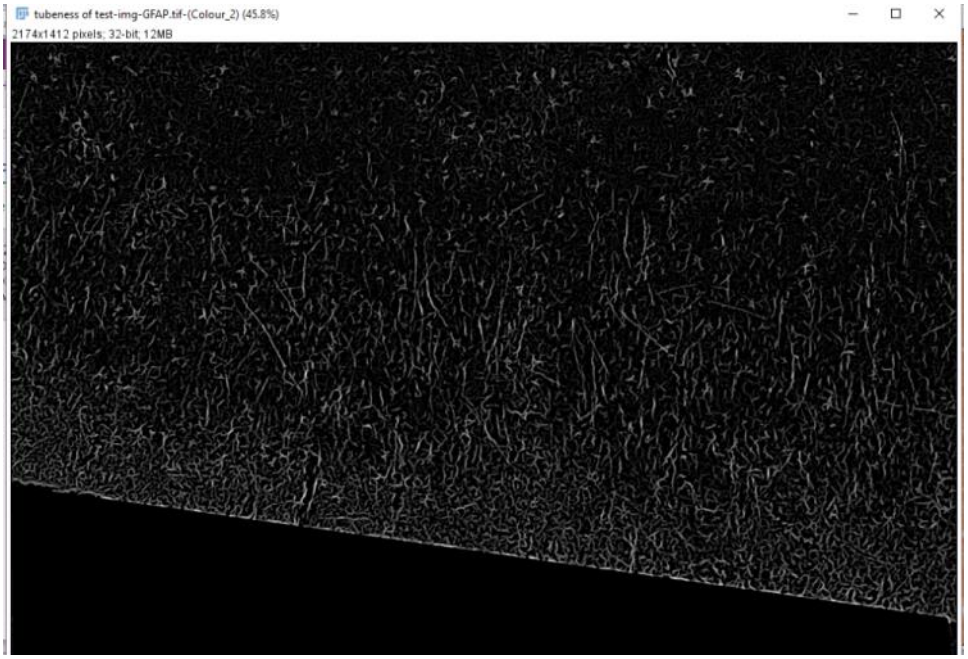


So, now, we can see that indeed the "tubes" are of the bright value.

Let's move to the procedure:
We open the command from **Plugins -> A:nalyze -> Tubeness**
We can now see the following window:



Before calculating the Hessian matrix for each point, the entire images is convolved with a Gaussian with the **standard deviation (sigma)** that you specify here. You can specify larger values of sigma here to tune the filter to select thicker tubes. Press "OK" and…



Some more things about Tubeness:
https://www.longair.net/edinburgh/imagej/tubeness/

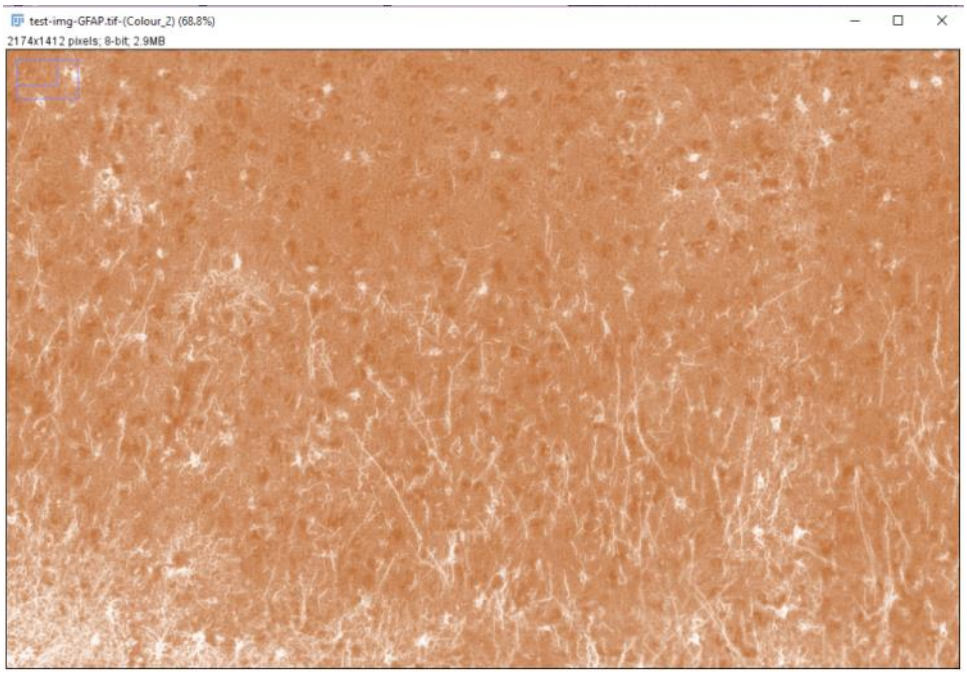
2017-03-23 (1)
FRANGI VESSELNESS

Τετάρτη, 22 Μαρτίου 2017 6:02 μμ

FRANGI VESSELNESS PLUGIN

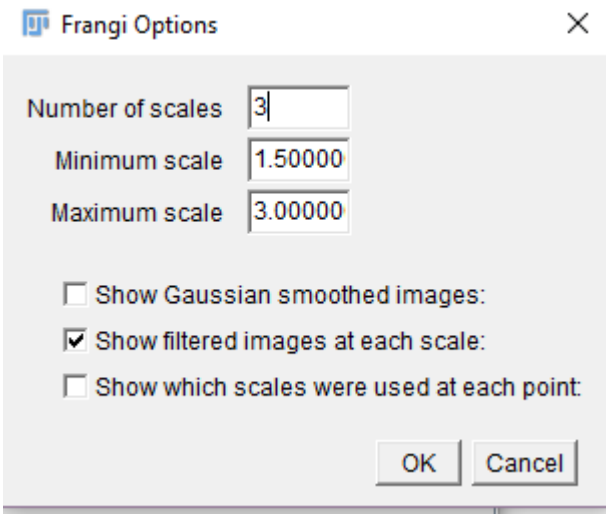
The Frangi Vesselness plugin is of the same function with Tubeness plugin. Though, I would say I prefer it, once it gives more controlled results, closer to the ones we desire. First of all we have to remember the important note of the tubeness plugin:

Once we need to distinguish the "tubes" we have to be aware that the regions of interest within the image HAVE TO BE bright (because this is what the plugin is looking to measure). If we run the "**look up table (LUT)**" we can see that the bright values are in fact the background of what we want to measure and not the neurons of which the structure we want to measure (which are the dark values). What we have to do is "**invert**" (inverts the bright and dark values of the image) **"Edit -> Invert**". So our new image of interest is the following:



So, now, we can see that indeed the "tubes" are of the bright value.

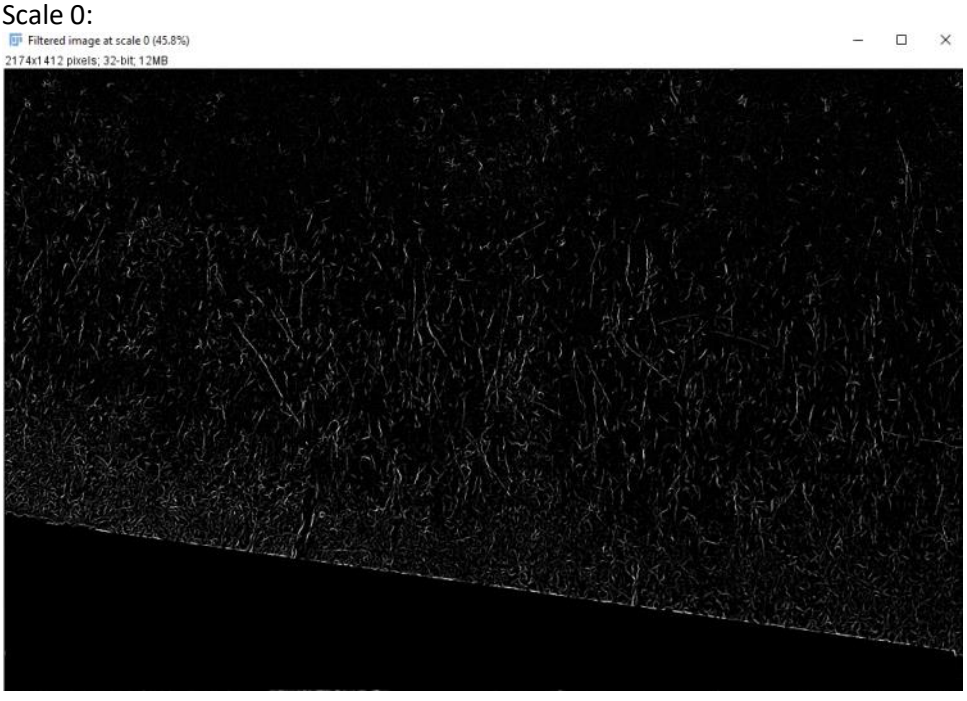
Now, we can run Frangi **(Plugins -> Process -> Frangi Vesselness).** Here, we can see that we have some more choices that we can make before we apply the settings to our image and they are shown in the following window:



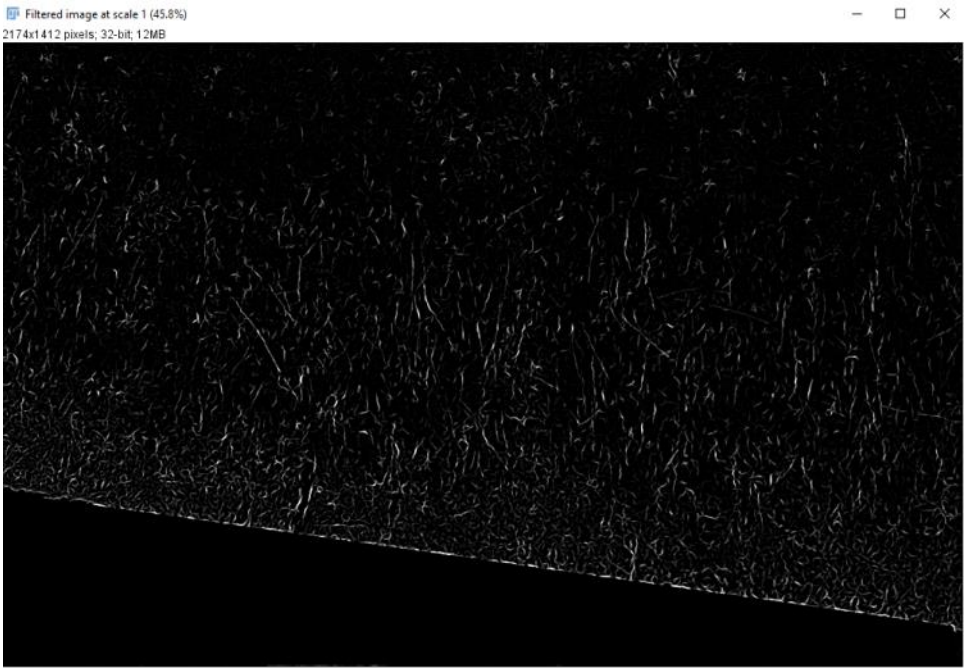
In order to get familiar with the sections within the window we have to be aware of what is "**Scales**" and "**Gaussian**" filter values.
Gaussian filter is the **value of blurring** which can be used in order to smoothen the background and have the noise of really thin structures cleared in order for the calculation of the desired thickness of the vessels to be done more precisely. (try this filter on sample image "Lena" within FIJI).
Scales is a value that needs to be set in order to get a better idea of the noise amount in various images. Depending on how thick the vessels that we want to calculate in the image, we can set different numbers of scales. In our case, vessels are not too thick but, for sure, they have an average thickness. Though, there are much more thinner structures that we cannot separate with bare eye, which is a noise that we want to avoid.
So, let's say that we will use 3 scales of a minimum scale of 1.5000 and a maximum scale of 3.000. (entered values in the above image).

We clicked the box "show filtered images at each scale, which means that besides the final image that occurs we also get three more images in order to get a better idea of the image at each value of the three selected scales we entered. The final image is a combination of the three scales from minimum to maximum scaling value we entered. So we have 4 images at last.

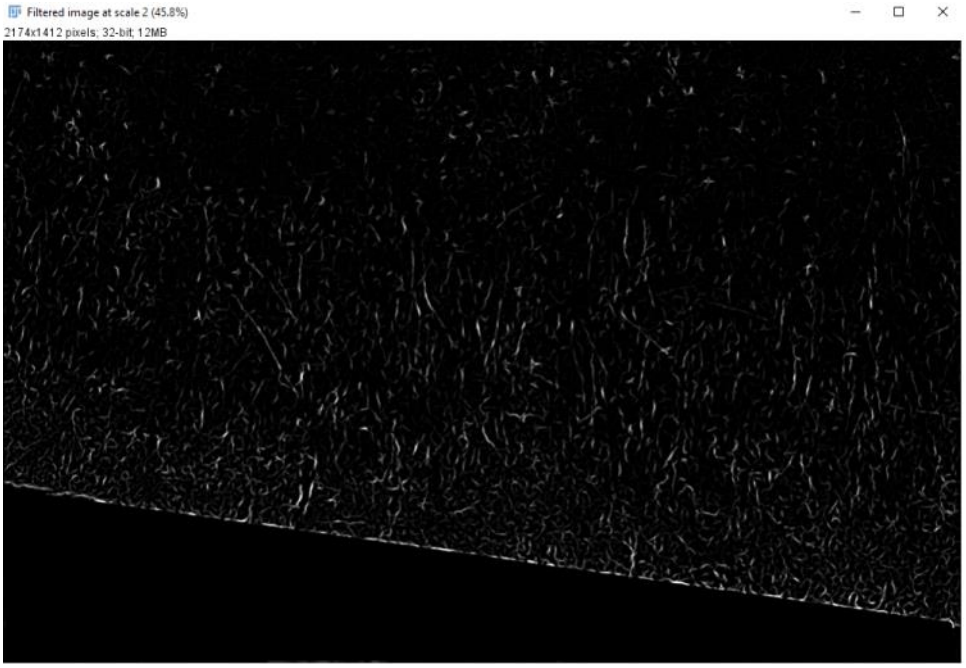
Scale 0:



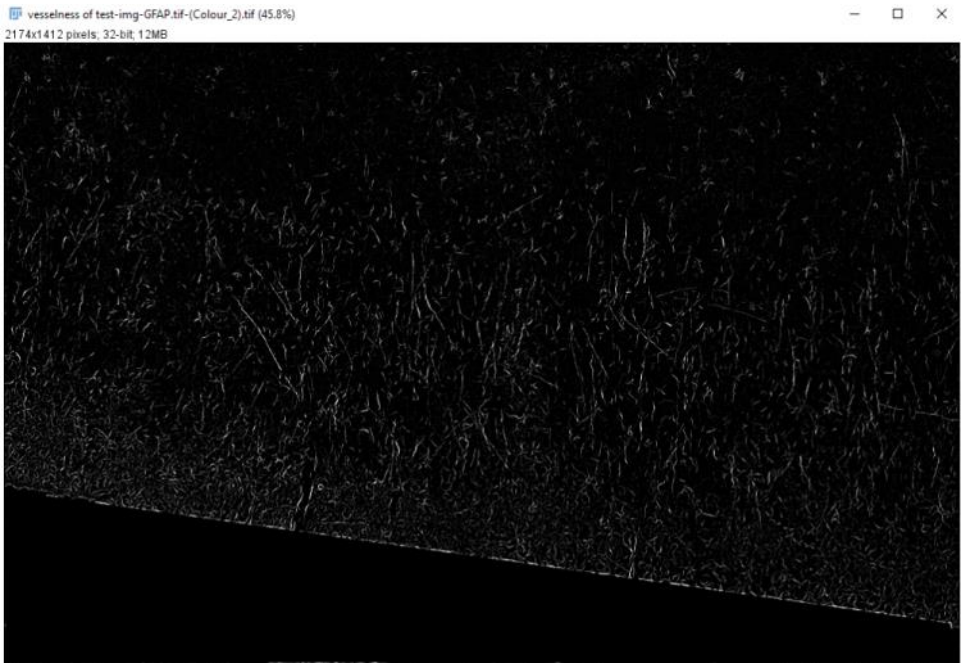
Scale 1:



Scale 2:



Final Image (combined scales from above):



http://imagej.net/Frangi

download

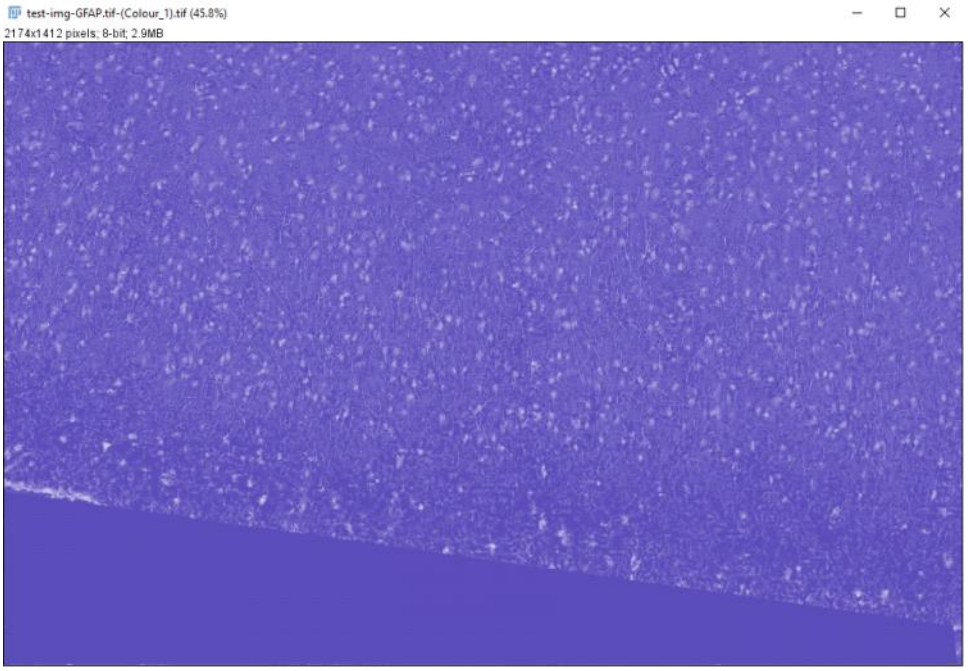
2017-03-23 (2)
Cell counting (with Trackmate)

Τετάρτη, 22 Μαρτίου 2017 6:05 μμ

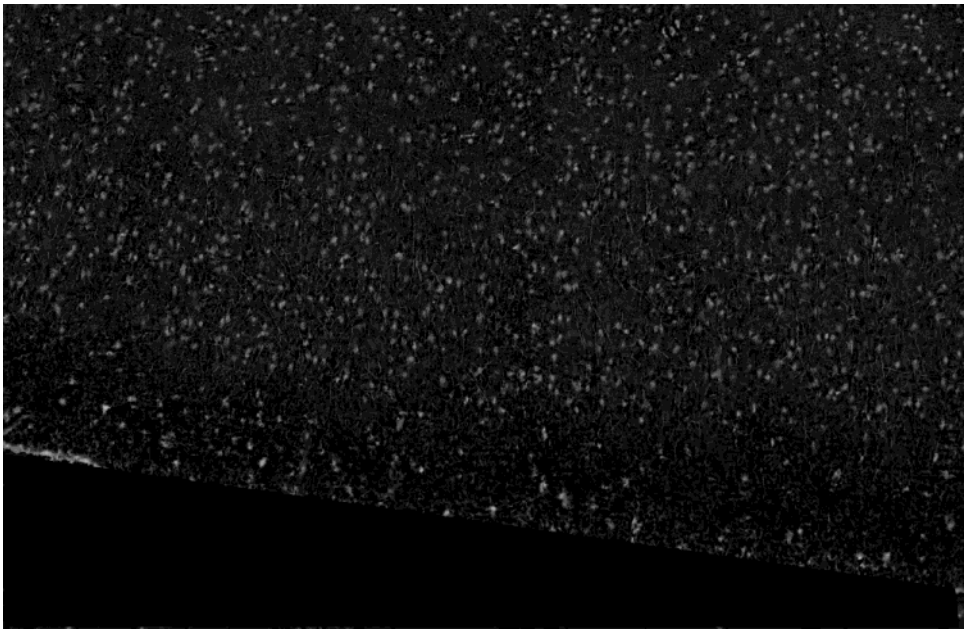
TRACKMATE PLUGIN

With the tester image that we have it has been kind of difficult to quantify the cells of the blue separated image that we created with Color Deconvolution.
The closest to success way to achieve that is Trackmate plugin.
First of all we have to invert the image in order for the cells (or blobs as seen with bare eye) to be bright. It is important that the threshold of the background is darker.

The inverted image is the following:

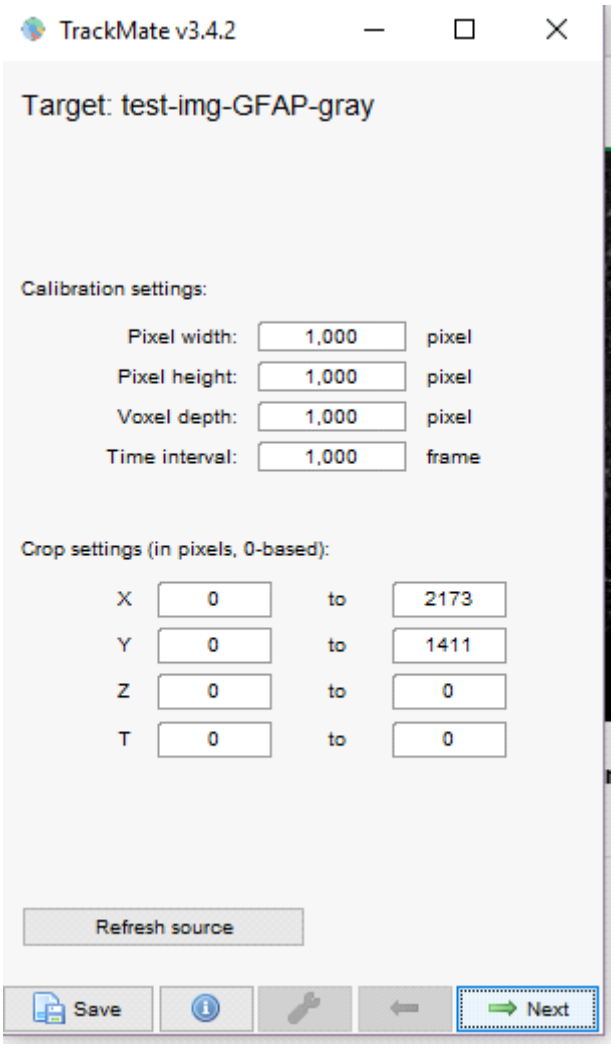


Now that the blobs are bright we could proceed to Trackmate but one more change would be helpful for both human and software to analyze the image. That would be greyscale conversion. So this image is created:

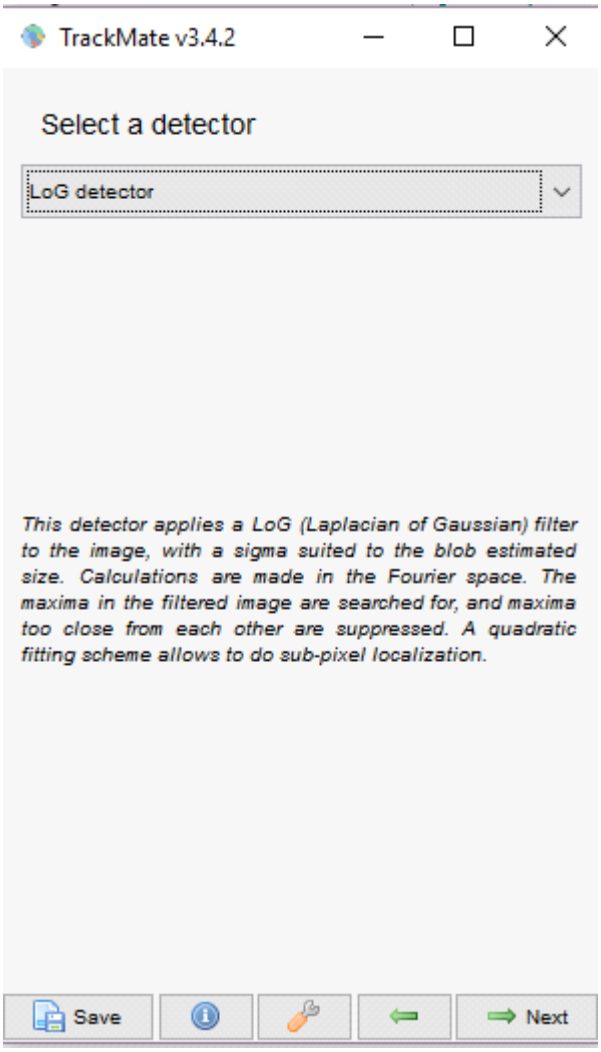


Now we can proceed with Trackmate **(Plugins -> Tracking -> Trackmate).** We get the following

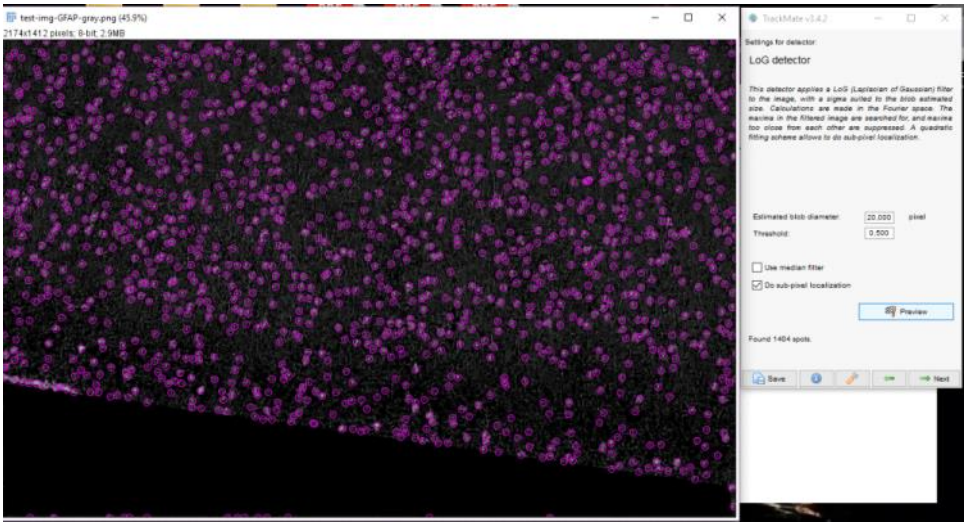
window:



We ignore the values (for the time being at least) and we go to the next window which is:



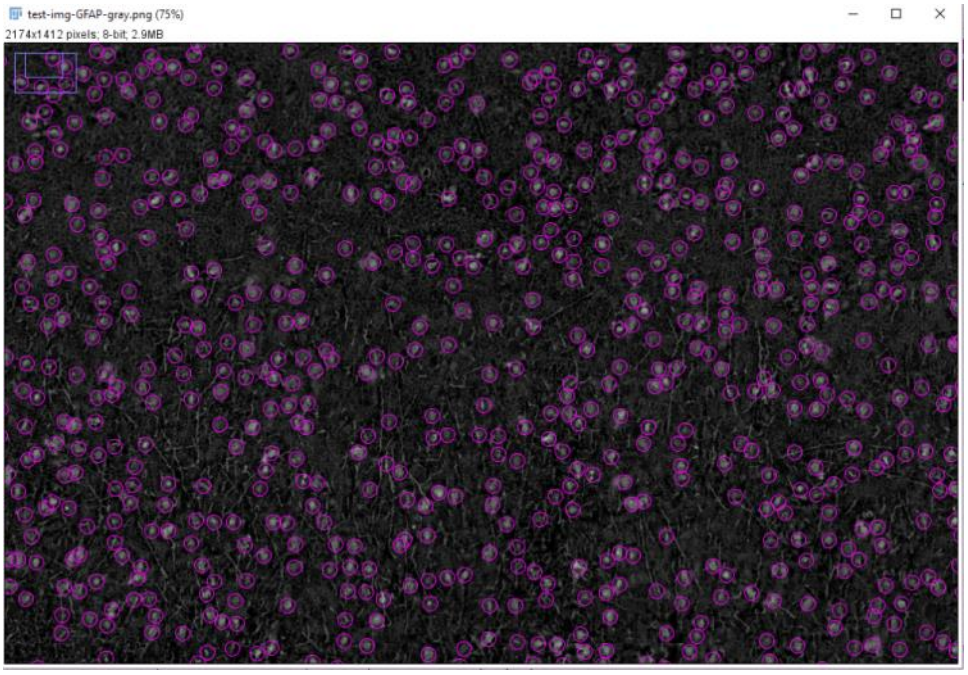
LoG (Laplacian of Guassian) detector is the best choice for our purpose.
We continue to the next window which is the most crucial:



This is the most crucial part because it is the dialogue window where values of (estimated) blob diameter and threshold have to be defined. A little it of playing needs to be done here once we have to understand the approximate diameter of the cell and of course the threshold once it is an important part for the background noise to be avoided (only if we adjust the value numbers at the window). (we have to press **"Preview"** for the left window to appear)

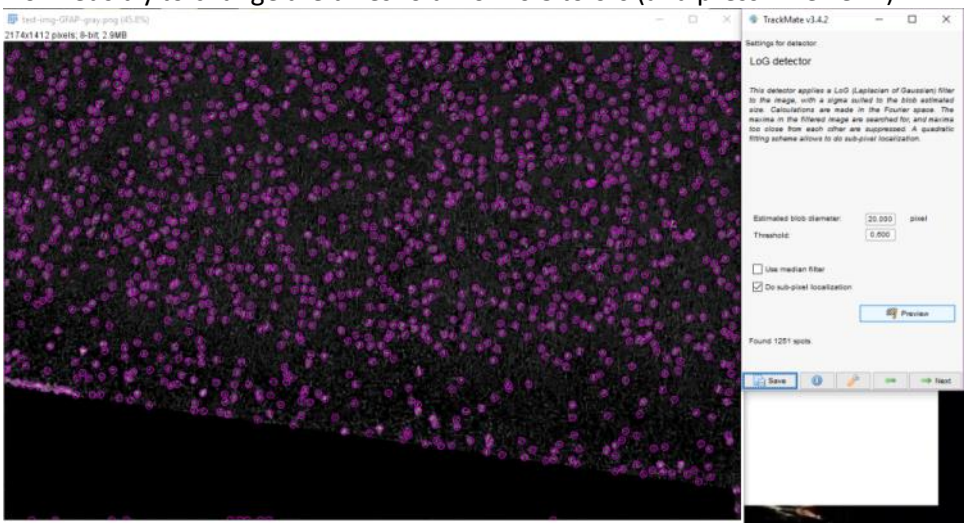
If we take a closer look at the right window, we can see that the dots that were detected are 1404 (having used 20 for diameter and 0.5 for threshold).
Let's zoom in the image and see if we can understand whether the measurement is approximate or

not:



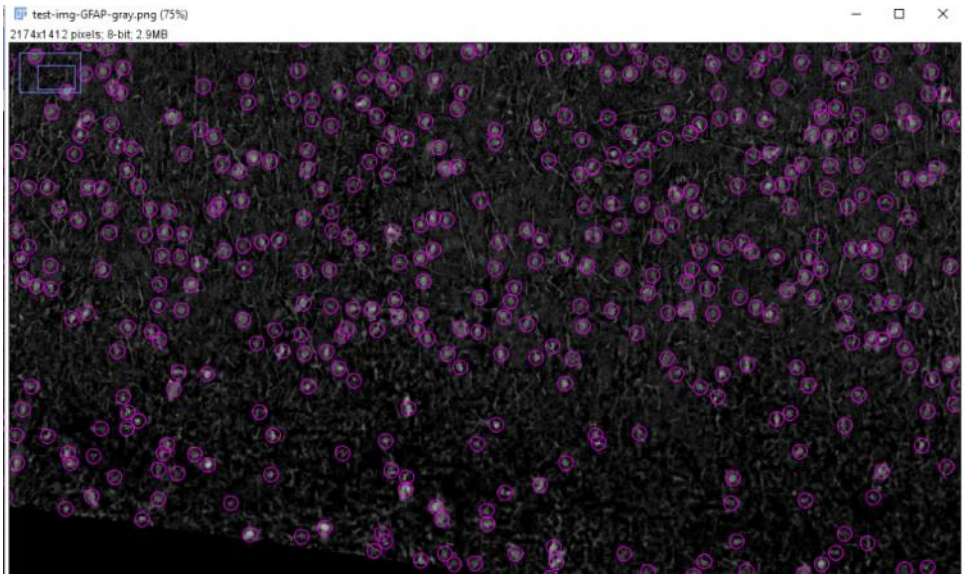
Seems to be pretty accurate.

Now let's try to change the threshold from 0.5 to 0.6 (and press "Preview"):



1251 spots detected.

And let's zoom in:



What is left is to understand is which threshold and diameter better fit the image. But as this is just a test image and not the whole sample that we examine, it needs some time in order to adjust the parameters to each occasion.

2017-03-24
Cell counting (another way)

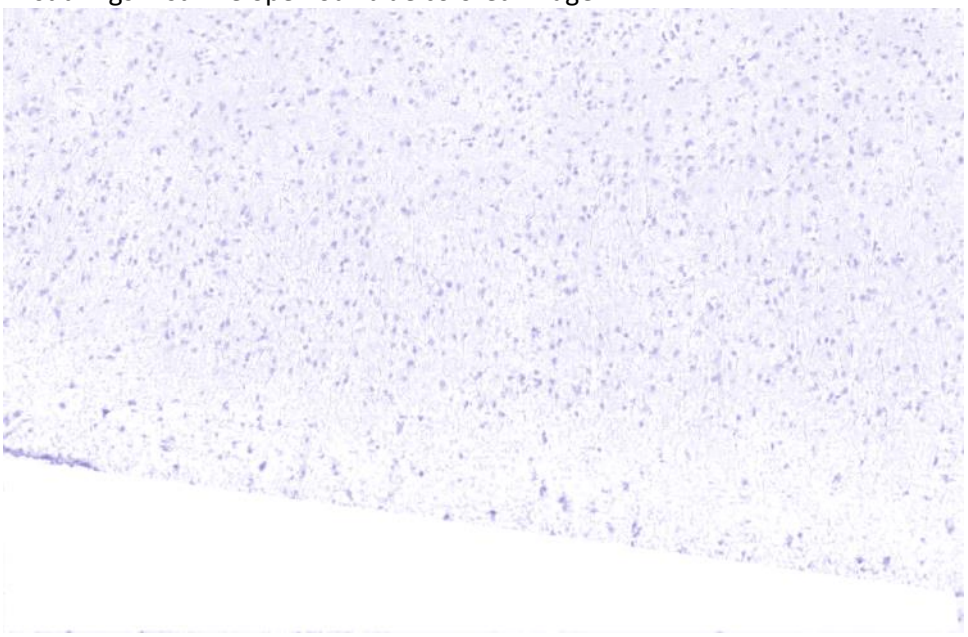
Πέμπτη, 23 Μαρτίου 2017 8:13 μμ

TRACKMATE #2

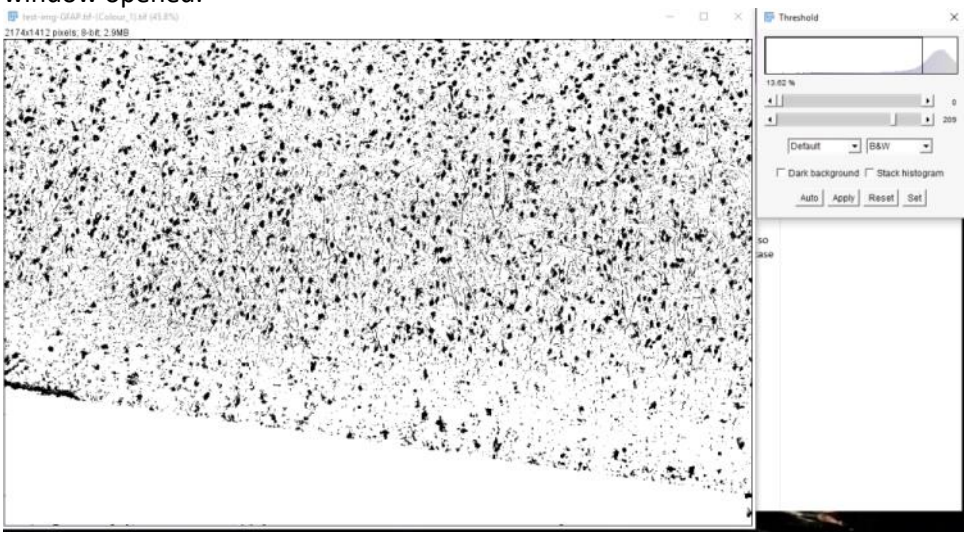
A way to make a quantitative analysis of the cells is with blob detection via Trackmate (as described above).

There can be another way, though.
What we can do is **thresholding** the image in order to have the foreground seperated from the background (to be brighter which is something that helps by means of counting). Then we have to **reduce the noise** of the background. This, as expected, will help us have a clearer image of our cells.

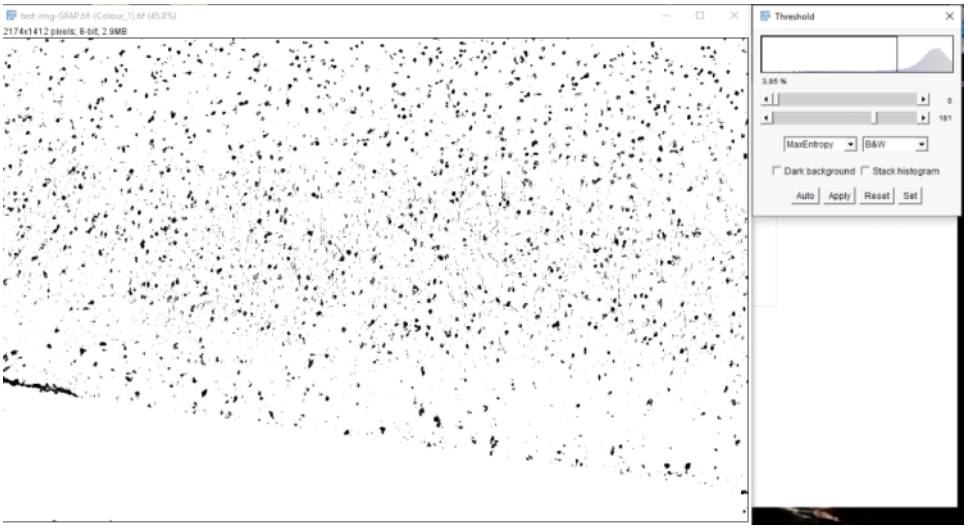
First things first. We open our blue colored image:



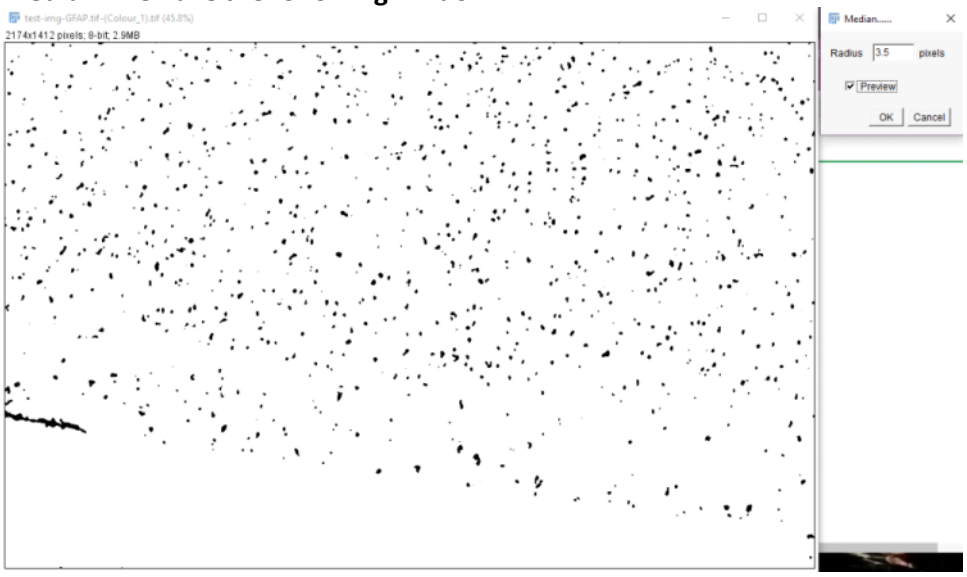
Right after, we have to threshold it. We go to **"Image -> Adjust -> Threshold"** and we have the right window opened:



Along with the right window, where values have to be input and/or filters to be chosen we see that the image has already changed. We are not quite satisfied by the default settings, though, so we have to apply other changes in order for the image to be as less "noisy" as possible. In this case the filter **"MaxEntropy"** seems to work better than the rest. So the final changed image is the following:



For sure, the background noise is still quite vast. What we can do is try to decrease it more.
A really useful tool to help us achieve that is the **"Median Filter". By going to "Process -> Filters -> Median" we have the following window:**



First, we choose **"Preview".** It will save us a lot of time from changes that cannot be undone after we apply the filter. We can play a little bit with the Radius pixels value in order to adjust the image the way we want it. In our case 3.5-4.0 seems to work fine. As shown in the image above, the background is much cleaner and we can proceed to particle quantitative evaluation. Which can be done with many tools within FIJI.

2017-03-28
Original Sized Samples

Τρίτη, 28 Μαρτίου 2017 5:43 μμ

We have 15 files of original quality to apply some settings on in order to find things we are interested in.

When we try to **open an image** with FIJI we have to choose among various recommended resolutions. We pick the second highest and then we have to be aware that we have to convert this image to an RGB image we can achieve by selecting **"Image -> Type -> RGB Color".** (we also have to unselect the "8bit" option).

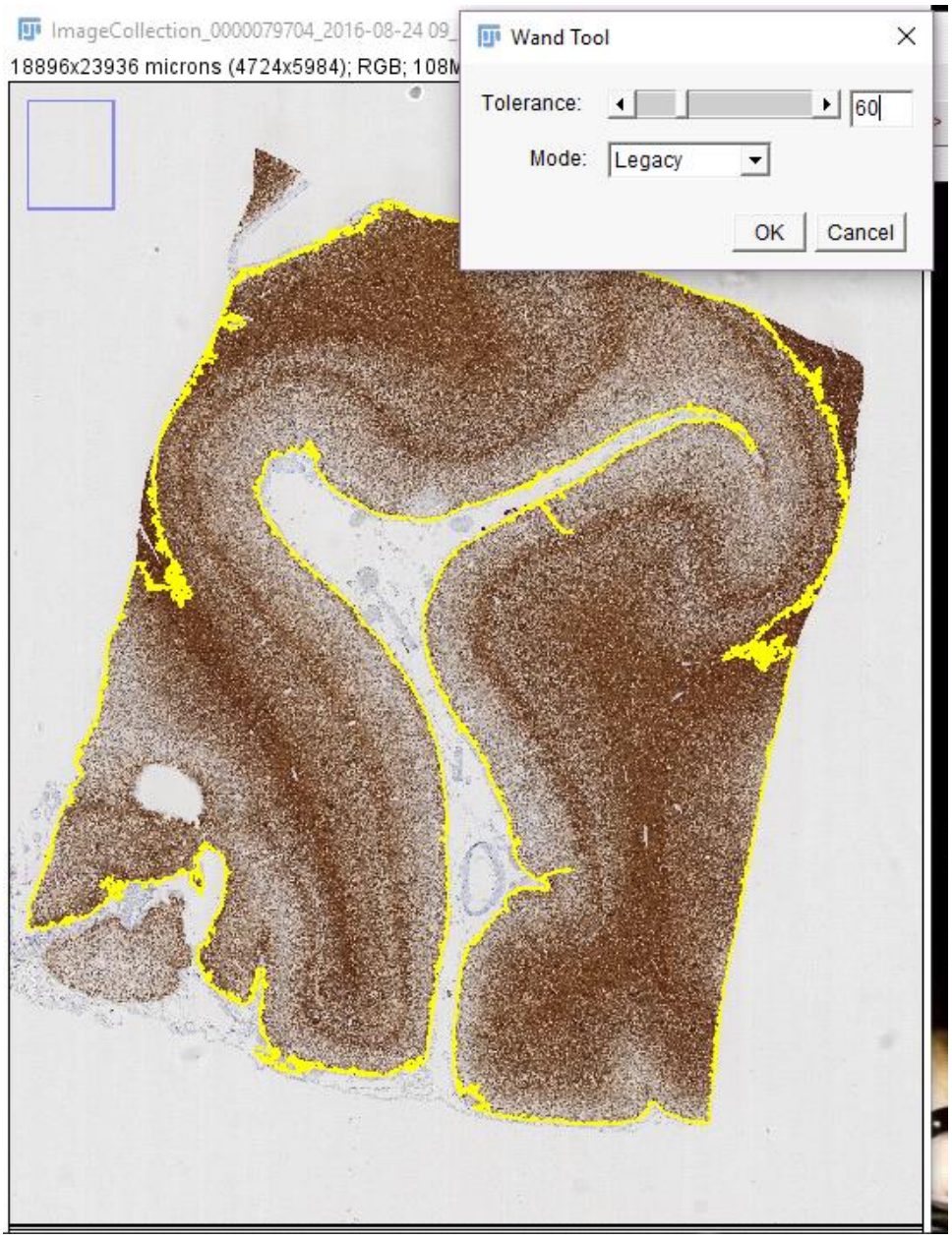
2017-03-28
Talk about Samples

Τετάρτη, 29 Μαρτίου 2017 3:09 μμ

ROI MARKING

We have to know the exact staining methods of each sample. By knowing this we will be able to know what we are to detect in each one of them . Also we will figure out which methods we are going to follow in order to reach the results we want (ie. Cell number, neuronal density etc).

One thing I tried was to create some ROIs beginning from the image with myelin density staining (i.e. SMI-94). Sure we can select the ROI by drawing it, but the automated "Wand" tool of drawing the borders is easier, for sure. So, the only thing we have to do it is select the wand tool and then play around by clicking to bordering areas of the image. The other thing we had to play with was the "Tolerance" box where we have to insert a numeric value. For the specific image the best would be 60.



To clear the background which would create noise and we don't need we select **"Edit -> Clear Outside"**

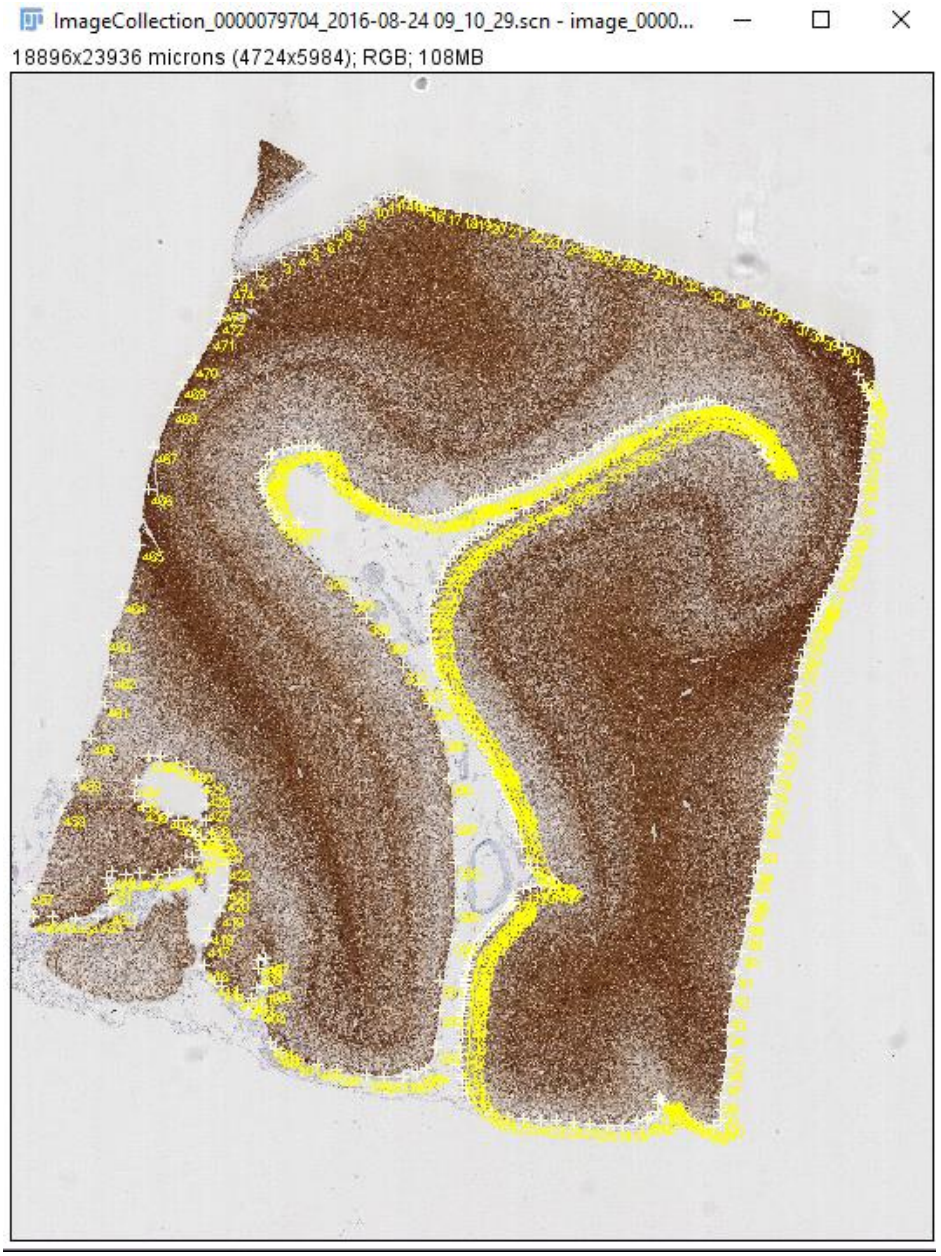

ImageCollection_0000079704_2016-08-24 0...
18896x23936 microns (4724x5984); RGB; 108MB

2017-03-30
Multi Point ROI

Τετάρτη, 29 Μαρτίου 2017 3:09 μμ

MULTIPOINT ROI

The tool with which we try to mark our ROI is "Multi-Point" (directly on the tool bar of FIJI) and we try to create dots in a logical numerical order (by shaping that makes sense by means of direction). After that we will interpolate the image in order to get the selected area and exclude the background we don't need.



Unfortunately, LiveWire (which is similar to multi-point but more automated than that, does not seem to install in fiji.

2017-03-31
Creating Macros

Σάββατο, 1 Απριλίου 2017 1:39 μμ

CREATING MACROS

Creating Macros has the difficulty of generating the code in order for the software to import all the images we want to process from a specific folder no matter how they are named and storing them into a specific folder under their new names created by the process.

The first step is to record a Macro for a procedure (eg Color Deconvolution, Tubeness, Frangi Vesselness). We have to be specific while recording it. Accuracy is very important. So we need to be processing our test image step by step and when the procedure is finished, to close the windows. We also want this to be recorded because if we have too many images to process we have to be sure that by the time the process of the Macro we created is over we will not have windows opened (because they can be many)!

Right after to modify the txt file in order for the Macro to be more specific. Particularly, we need to give to the Macro a description of the procedure that it has to make. Right after, we have to generate the input (and if necessary) output folder. Then we paste the text of the recorded macro we created (some modifications by means of image titles and output folder have to be also done there) and then we have to set the input folder in another line. In the end, we need to create a variable list for the sequence of the dataset (of the total number of images that will be proceeded). We save the file and we can run it as a macro from the plugins option of fiji.

Useful link for the above:
http://imagej.net/How_to_apply_a_common_operation_to_a_complete_directory

2017-04-07
Trainable Weka Segmentation

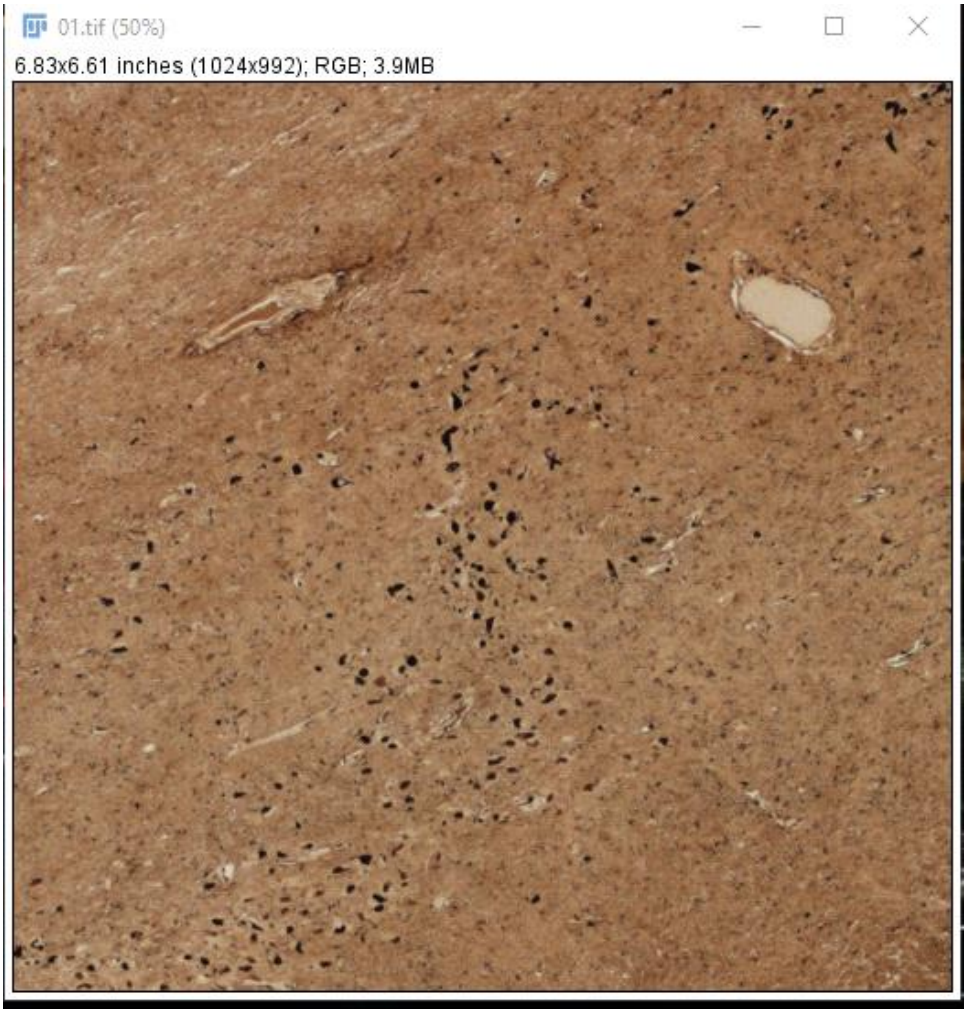
Παρασκευή, 7 Απριλίου 2017 2:53 μμ

TRAINABLE WEKA SEGMENTATION

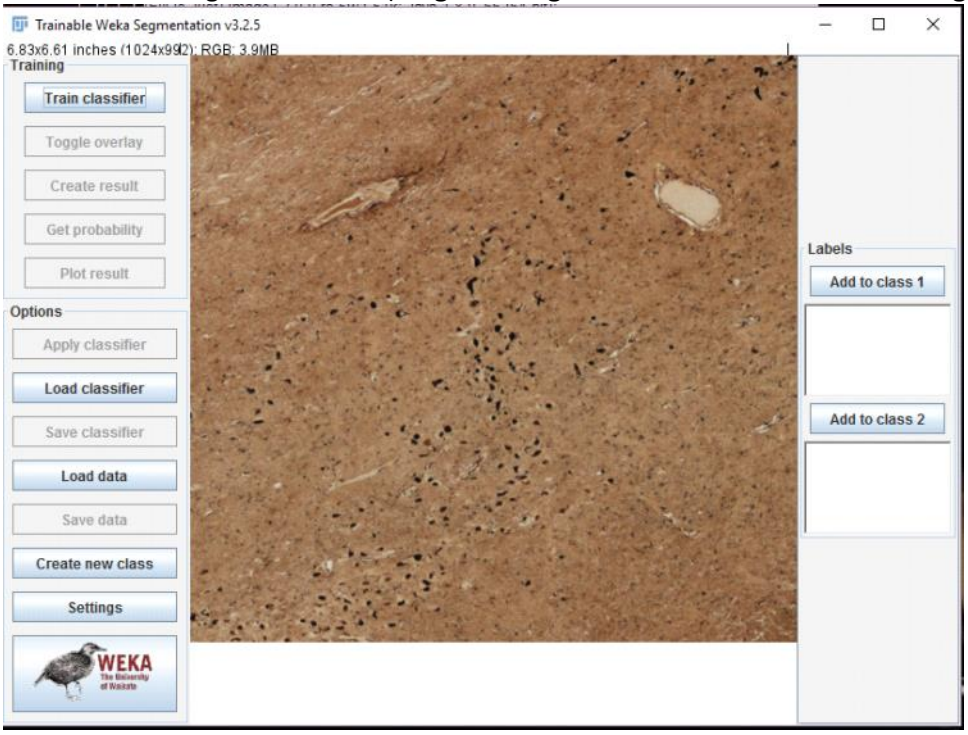
This is a really nice way of clearing out areas that we are interested in by training the software in order to be recognizing as foreground only the marks that we really want and no irrelevant noise.

Let's say we have the following image of which we want to get the number of neurons in the white matter.

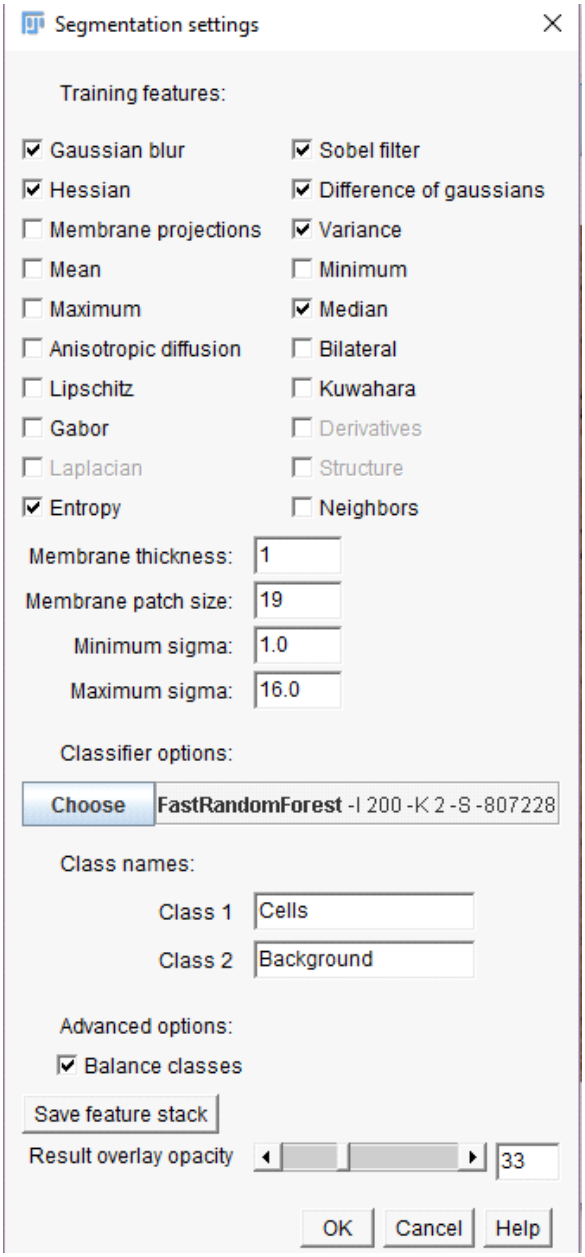
This is a really big one. We don't need most of it. So we would only crop the white matter in order to start the process. Though, since it requires some (ok a lot of) time we will go on with a tester.
So, we take a closer look in the image and we crop just a small part of it.



We only need the black spots (the neurons). How do we separate the back from the fore ground? Let's start training our software (**Plugins -> Segmentation -> Trainable Weka Segmentation**):



A quick overview of the settings… There are a lot of options which would help the program detect and distinguish the two areas we are interested in (foreground and background) but we need to use specific ones depending on the kind of picture that we have. So, in this case that would be these settings:



Now, before we press the option "Train Classifier" to start the procedure, we need to add the areas we want to separate from each other in the classes boxes. We will do it with the default selection tools of FIJI (circle, rectangular, freehand etc) and we will be adding the cell marked areas in class 1 and the background areas in class 2. So, we will have the following result:

Just have a look at the green and red marked areas. We are ready to go with Train classifier now.

After the classifier After the classifier gives us the result we can see the image in the window separated in two classes of green and red color. We will save the classifier to a folder and then we can apply it to any image that contains that area in which we are interested in. After the filter applies to the image we can get the probability map. It shows us an insight of how probable is something to be the structure we are interested in (since it gives us two stacks we can just remove one by going to

*Install ImageScience Library
http://imagej.net/Trainable_Weka_Segmentation

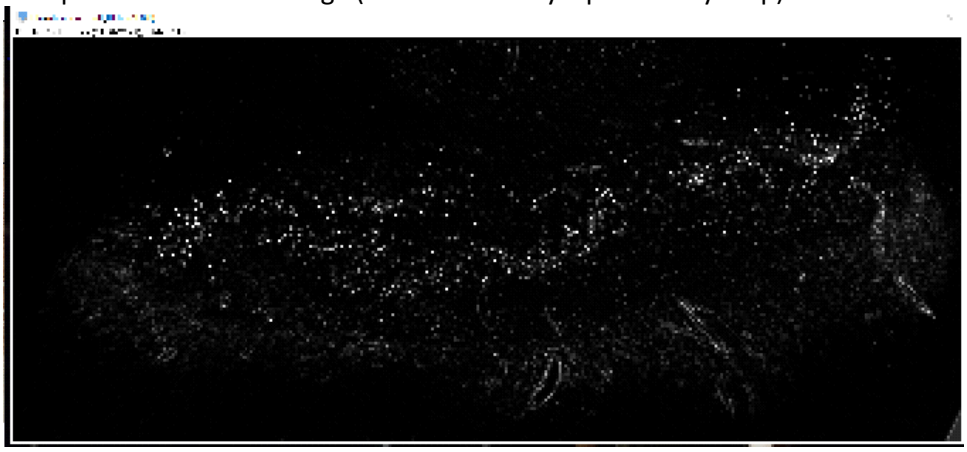
2017-04-13
IJPB Morph

Πέμπτη, 13 Απριλίου 2017 2:36 μμ

MORPHOLIBJ

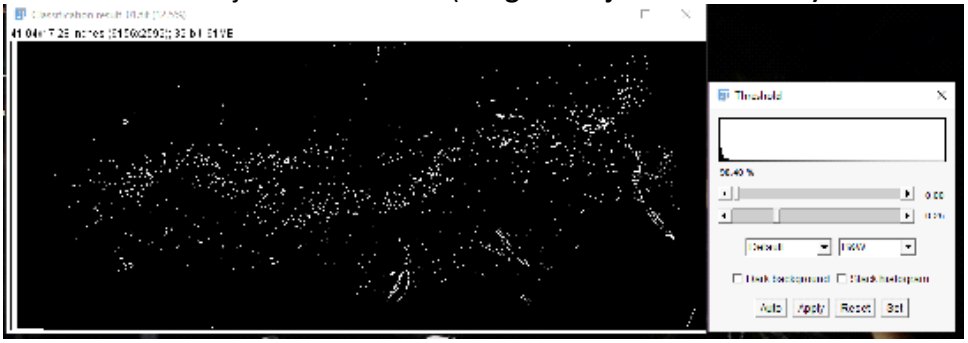
What would be a way to count cells after all? Along with that, what would be a way to measure distances, radii, thickness and other factors which we are interested in?
We will try MorphoLibj plugin. Without much introduction, we want to measure some things from this image:

we open our classified image (which is actually a probability map):



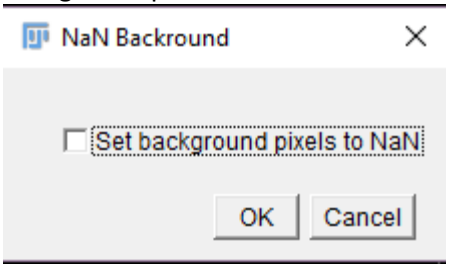
If there is a second image of the stack created after the classification we can delete it **(Image -> Stacks -> Delete Slice**).

Now we have to adjust the threshold (**Image -> Adjust -> Threshold):**

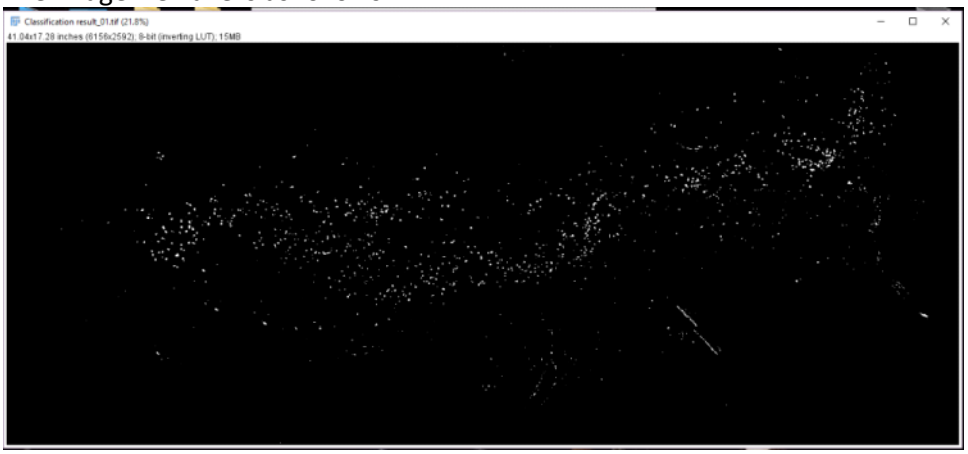


If we move the cursor on the white spots (which the program realizes as cells) we will see a value on the toolbar which claims the probability of this specific white spot to be one of the cells we are interested in or another similar structure or just a noise. In our case we will set the probability of the

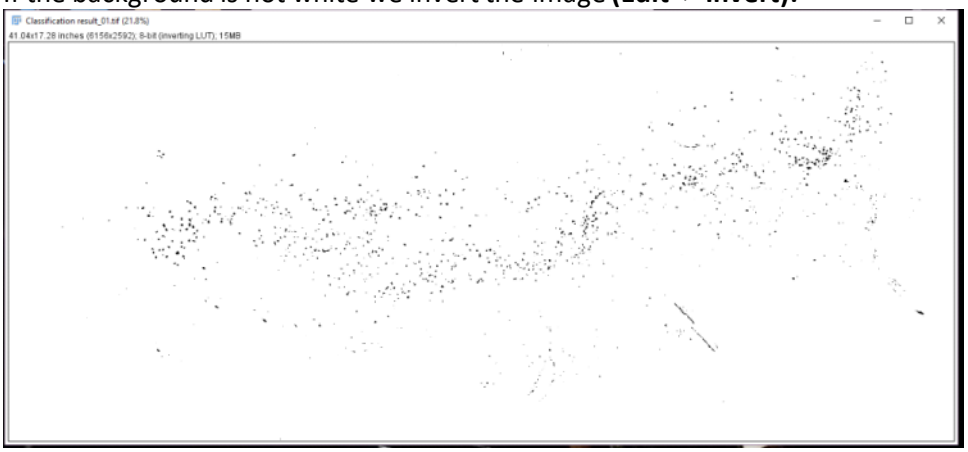
threshold at around 50%. When we press "apply" a window appears and asks if we want to "set background pixels to NaN". We don't want that so we uncheck:



The image we have is as follows:



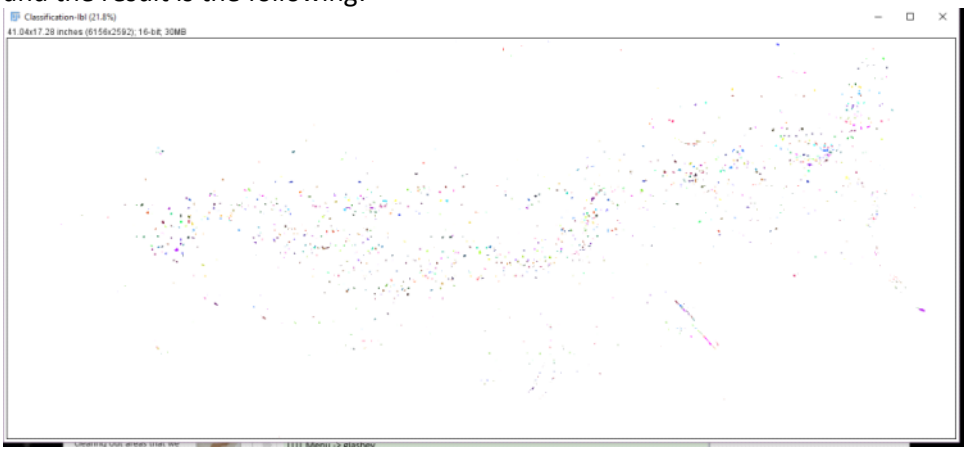
If the background is not white we invert the image **(Edit -> Invert):**



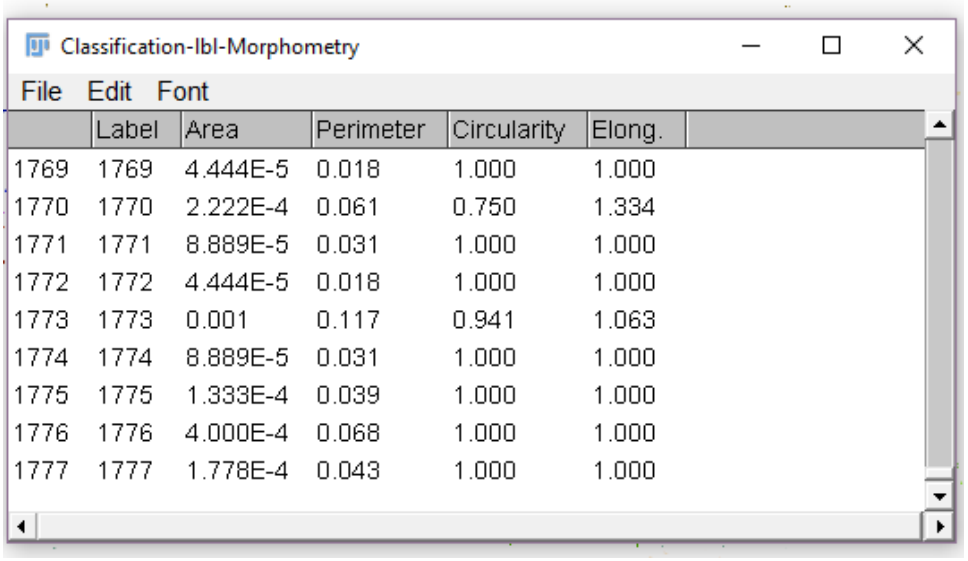
Then we have to watershed the image (automated, really fast procedure) which will separate cells that are fused and counted as one **(Process -> Binary -> Watershed).**

Now we will use the plugin **(Plugins -> MorphoLibJ -> Binary Images -> Connected Components Labeling).** This will differentiate out structures apart from each other. We will have a window on which we press "Ok" and then we have this image created:

The image above has all the cells separated but we need to take a better look so we can change the LUT with various filters. In this case we will use the Glasbey filter (from the LUT menu on the toolbar) and the result is the following:



We can tell by bae eye that our cells are separated. But how is this useful?
By using other tools of the plugin we can get much more information. If we go to **Plugins -> MorphoLibJ -> Analyze -> Region Morphometry** we will get a log file like the following:



Classification-lbl-Morphometry

File Edit Font

	Label	Area	Perimeter	Circularity	Elong.
1769	1769	4.444E-5	0.018	1.000	1.000
1770	1770	2.222E-4	0.061	0.750	1.334
1771	1771	8.889E-5	0.031	1.000	1.000
1772	1772	4.444E-5	0.018	1.000	1.000
1773	1773	0.001	0.117	0.941	1.063
1774	1774	8.889E-5	0.031	1.000	1.000
1775	1775	1.333E-4	0.039	1.000	1.000
1776	1776	4.000E-4	0.068	1.000	1.000
1777	1777	1.778E-4	0.043	1.000	1.000

Counting, Perimetres, area information of each cell and other parametres are shown.

(SHORTCUT:
Open classified image
Delete second stack slice
Threshold it (NO set background pixels to NaN)
Invert (if background not white)
Watershed
MorphoLibj ->binary images -> connected components labeling
LUT Menu -> glasbey
MorphoLibj -> Analyze -> Region Morphometry (and save log file)

http://imagej.net/MorphoLibJ)

2017-04-28
Trainable Weka Segmentation

Παρασκευή, 28 Απριλίου 2017 10:27 πμ

FEATURES IN TRAINABLE WEKA SEGMENTATION

In Trainable Weka Segmentation saving the classifier is different than saving the data of all the markings you created.

Each feature saves different types of information.

We can apply the data information to a stack at the 2D segmentation (but not at the 3D).

For the 3D we would have to create a classifier of all the combined data from the 2D training and then apply it to the 3D sample.

The filters that we use for the training of the data of the 2D Segmentation are:

- Gaussian
- Hessian
- Anisotropic
- Gabor
- Variance
- Structure
- Neighbors
- Laplacian

And the thickness should be set to 5 (in this case… in general we would have to find out the proper thickness).

2017-05-16
Ilastik

Τρίτη, 16 Μαΐου 2017 12:45 μμ

ILASTIK

With Ilastik we can train classifiers on raw data by means of object size, shape and other quantitative parameters. There are various processes that we have to apply given step by step by the software within the interface.

Then, we can label our objects of interest and create probability maps. We can export the data and we can also apply the settings to other raw data of interest. We can even save the Project that we have created and when we load it all the settings will still be applied and ready to process from the spot we have saved the whole procedure.

The most important functions that we can use are "Pixel Probabilities" and "Object Probabilities". We can generate and export all probability maps of our interest by saving them in many different types.
We can also batch the procedure to other samples. Using the same filters that we used to process the tester and after we are satisfied by the result we can apply it in larger images and then we will be able to have the whole probability map for the dataset.

http://ilastik.org/documentation/index.html

2017-05-31
Directionality (ImageJ)

Τετάρτη, 31 Μαΐου 2017 3:14 μμ

DIRECTIONALITY

https://imagej.net/Directionality

2017-05-31 (2)
OrientationJ

Τετάρτη, 31 Μαΐου 2017 5:40 μμ

ORIENTATIONJ

1) **OrientationJ Analysis:**

A) Color Survey
Hue: Orientation
Saturation: Coherency
Brightness: Original-Image

We can split our image in different other images showing the orientation, energy, color survey, coherency and so on.

http://bigwww.epfl.ch/demo/orientation/

2017-06-19
Bodian sample probability maps,
Voxel splitting,
Fiber orientation histograms

Δευτέρα, 19 Ιουνίου 2017 3:50 μμ

For the Bodian stain which shows us the myelin density of the sample we trained a cropped image inilastik software. After we tried to create probability maps for this testing image under diferent pixel values (3,5, 5 and 10), we batched the trainer to the whole original image. ImageJ cannot process this whole image, so we split it in two parts with the help of Matlab. Then for each different pixel value probability map we have two parts which are the ones we can actually process further.

Now that we have created the probability maps for the BODIAN stain image which shows the myelin density (with the help of ilastik software) we have to create histograms for the fiber orientation.

What we have to do in the first place is to split the probability map images in voxels with the help of a macro which requires us to input the value of pixels that we desire and a destination folder for the voxels that will be created.

After that, we have to apply another macro to the folder with the splitted voxels which will create a histogram for each file separately. The only thing we have to insert is an input and an output folder.