# Array tomography Matlab tool protocol

## 0. Previous steps.

- To run the following tools, the images should be saved as stacks of consecutive images, each channel separated and correctly named (see note). Whole images can be used (instead of small crops).

- If there are different cases or section with the same combination of channels, all can be processed together if they are located in the same folder (i.e. Case1\_syph.tif, Case1\_abeta.tif, Case1\_psd.tif, Case2\_syph.tif, Case2\_abeta.tif, Case2\_psd.tif …).

- **IMPORTANT NOTE**: try to keep the image stack names simple, if possible without spaces and without repeated sequence of letters (i.e. don’t use syph and syp, you could use syph and syp1). In many parts of this script we try to find an image based on its channel name, so we will load all the images with the channel name in the image name.

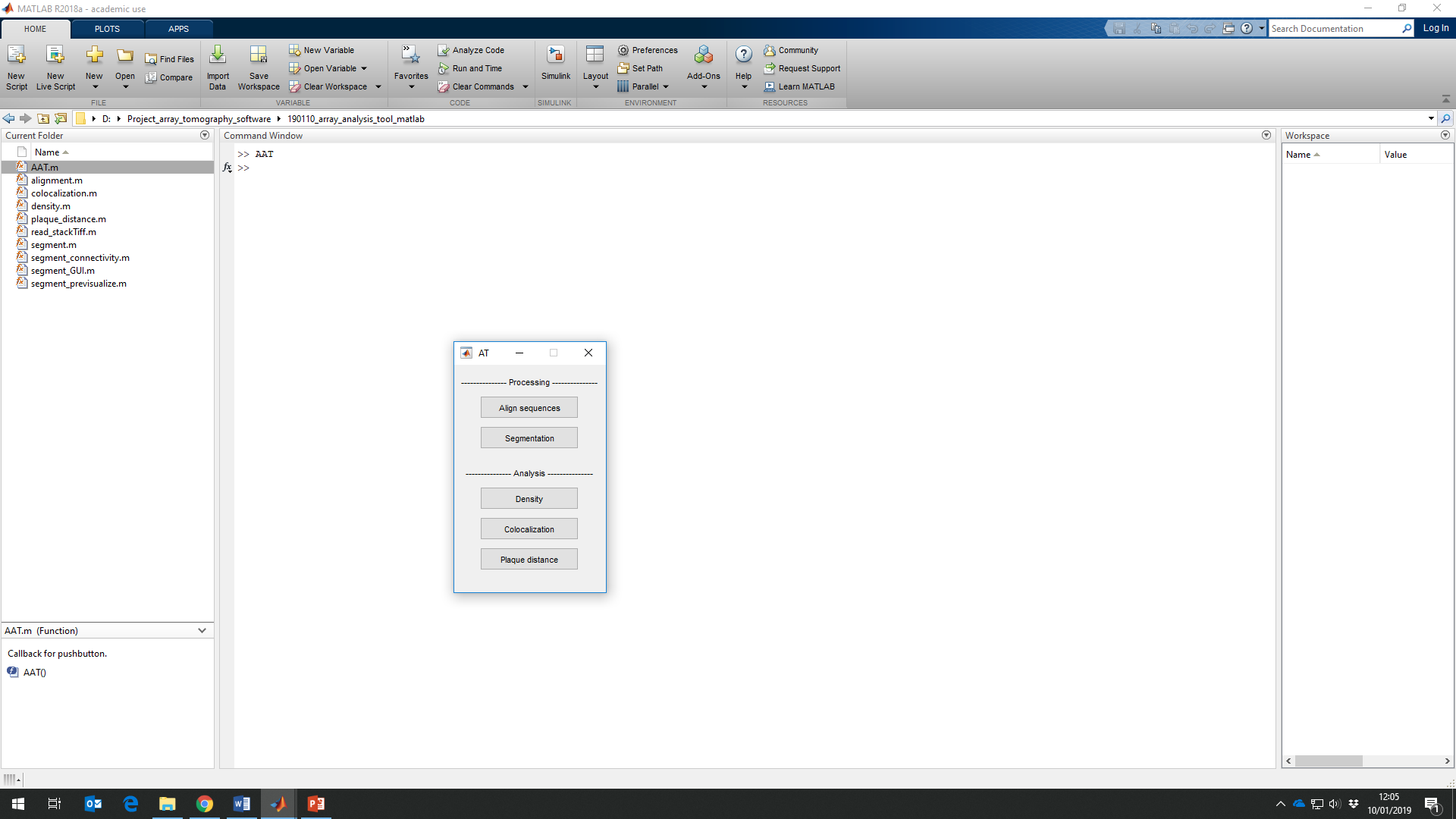
## 1. Run the script.

- Copy the script folder from the server to your computer.

Location: *\spiresjoneslab\protocols\array tomography protocols and software\*

Folder: ***190115\_array\_analysis\_tool\_matlab*** (check the date for last version)

- Open Matlab and find the folder.

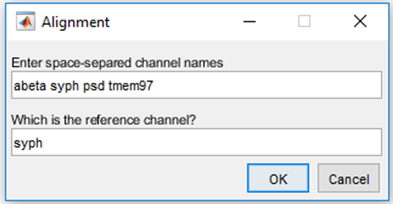
- Right click on **AAT.m** and press **Run**. The following menu should pop up.

## 2. Align sequences (registration).

With this tool you can align stacks of images.

- Click the **Align sequences** button on the main menu and select the **folder** where are located all the images to align.

- Write the name of the different channels to be aligned and choose one of them as the reference channel.

Reference channel should be the cleanest channel, with well-defined structures. Usually synaptic patterns such as PSD95 or Synaptophysin worked nicely.

- Press **OK** and wait. It can be long depending on the size of the images. You can follow the progress in the **Command Window** of Matlab.

- All the images will be saved in a folder called **Aligned**.

- Check the alignment using ImageJ. If images are not correctly aligned, try with other reference channels or process the images before with ImageJ (remove bright objects, clear background…).

## 3. Segmentation.

This tool allows for the isolation (segmentation) of 3D objects that will be analysed in the following steps. Each channel will need different parameters to be segmented (because of different shapes, fluorophores…). It means that as a previous step should be created independent folders for each channel (i.e. folder syph > Case1\_syph.tif, Case2\_tif…; folder psd > Case1\_psd.tif, Case2\_psd.tif…).

- Click the **Segmentation** button on the main menu.

- First select the folder where are located all the images of the first channel to segment pressing **Select Path (1)**. It will pop up a menu where are listed all the images in the folder **(2)**.

- Try different parameters to select the objects of interest **(3)**. This segmentation is based on an automatic local threshold made in a small window where a mean or median of intensities is performed. So the following parameters can be modified:

- **Window size**: Should be a wee bit bigger than the object of interest.

- **C factor**: Correction factor. Higher it is, more permissive you are and lower intensity objects are detected.

- **Mean** or **Median**: Usually mean works fine, there are not big differences between them.

- **Filt object size**: You can choose the min and maximum object size you want to detect, in pixels. Less than 3 is not possible. In the previsualization it is not shown.

- Press **Previsualize** to play with the parameters **(4)**. It will pop up two windows: the original image, and the segmented image. You can move through the stack to check the segmentation.

**IMPORTANT NOTE**: The previsualized images are a small section of the image (100x100 pixel) and the intensity of the original image is adjusted just for visualization purposes. In the segmented image, all those objects that are not found in at least 2 consecutive sections are removed.

- Some tips to decide the parameters:

- Move through the sections of the original image looking for objects that are present in consecutive images and check if they are present in the segmented image (in case you are losing objects).

- Move through the sections of the segmented image and check if the objects selected are present in consecutive images in the segmented image (in case you are taking too many objects).

- Try to find the parameters that better preserve the original shape of the object. In synapses or punctate objects, try to avoid two rounded elements to merge into one.

- Once you are happy with one image, check the parameters with the other images in the folder.

- The same protein stained with the same fluorophores usually requires the same parameters.

- You can perform the segmentation and watch the whole image in ImageJ comparing it with the original for a more accurate evaluation.

- If controls (without the channel) are included in the folder, may help deciding the c factor.

- The human eye is better than this algorithm, do not desperate.

- Once you are happy with the previsualized segmentation, press **Analyze (5)**. You can follow the progress in the **Command Window** of Matlab.

- The segmented images will be saved into a folder called **Segmented**.

## 4. Density.

This tool calculates the number of 3D objects present in a stack. Input images need to be already segmented.

- Click the **Density** button on the main menu and select the **folder** where are located all the images to analyze.

- Write the name of the different channels to be analysed and choose one of them for the neuropil area calculation.

The neuropil area calculation is made using a maximum intensity projection and a coarse segmentation. This will allow us to know the area without cell bodies and big blood vessels. For better performance, dot-like dense channels and long stacks are preferred.

- Press **OK**. You can follow the progress in the **Command Window** of Matlab.

- The densities and areas will be calculated and saved into **Density** Excel file inside a new **Results** folder.

Neuropil mask will be also saved into a folder called **Neuropilmask**.

## 5. Colocalization.

This tool allows the calculation of the percent of objects of a channel colocalization with another channel by distance or overlap. There is not a maximum number of channels, but keep in mind the number of combinations. Input images need to be already segmented.

**IMPORTANT NOTE:** In big images with many objects it can be a really long process (i.e. colocalization between 3 channels with ~10000 objects each lasts ~2h). If it is a big study, try with a single case first to approximate the total duration.

- Click the **Colocalization** button on the main menu and select the **folder** where are located all the images to analyze.

- In the menu that pops up, introduce:

- Name of the different channels to be analysed.

- Resolution of the images (used for the distance).

- Maximum distance between objects to consider colocalization. If not using distance leave it by deafault.

- Minimum overlap between objects to consider colocalization. The area taken into account is the main object area.

- Press **OK**. Then it will ask the desired **type of colocalization** between each pair of channels. More channels, more combinations.

- Finally it will ask if you want to **save the images** of the colocalization. If **Aye**, it will save, for each channel, all the possible combinations (i.e. for abeta, psd and syph colocalization it will save an image of objects abeta+abeta, abeta+syph, abeta+psd and abeta+syph+psd; syph+syph, syph+abeta…).

- You can follow the progress in the **Command Window** of Matlab.

- In the **Results** folder you will find a **Colocalization** Excel file with the percent of object of each channels in each condition as well as the number of objects found.

It also save a **.mat** file that have all the information generated in the script. It may be useful if you want to come back and use i.e. the exact distance between the objects.

If **Aye** was selected, the images with colocalizing objects will be saved in **ColocImages** folder.

## 5. Plaque distance.

This tool outputs the number of objects found at a certain distance of the perimeter of an amyloid plaque. Input images need to be already segmented. If the question is related with colocalization, the images generated in the colocalization tool can be used (again be careful with file names).

- Click the **Plaque distance** button on the main menu and select the **folder** where are located all the images to analyze.

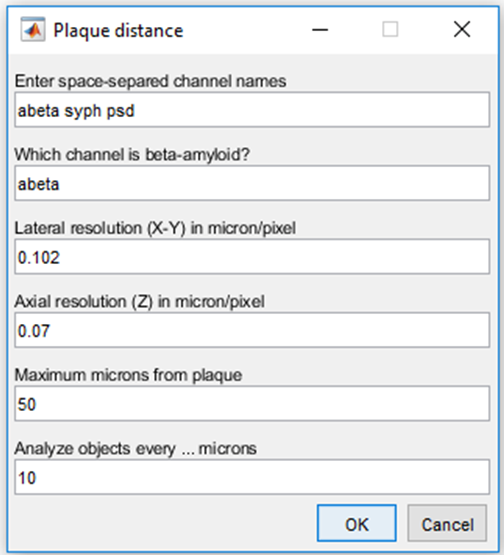
- In the menu that pops up, introduce:

- Name of the different channels to be analysed.

- Name of the beta-amyloid plaque channel. 6e10 cannot be used (looks exponential).

- Resolution of the images (used for the distance from plaque).

- Maximum distance between objects and plaque perimeter.

- The distance where objects will be analysed together. Divided to the maximum microns will give us the number of resulting bins.

- Finally it will ask if you want to **save the images** of the bins. If **Aye**, it will save, for each channel, the objects located at each bin (every x microns decided before).

- Press **Ok** and follow the progress in the **Command Window** of Matlab.

- In the **Results** folder you will find a **Plaque\_distance** Excel file with the number of objects of each channel that are present in each bin surrounding the plaque.

In **Masked\_images** folder are saved the mask of the plaque and all the channels with this mask applied.

In **Bins\_images** folder are saved an image of the boundary used for the analysis and, if selected, all the images of each bin.