1. Select the best images from each animal and copy them to a different folder (Folder A). Specifically for this step you can *:*
   1. Isolate channel of interest with Separate\_default\_not\_pickle.ipynb for all images

(or, alternatively, do it with your own macro using Fiji).

* 1. Create a different folder (Folder B) and from Fiji -> Process -> Batch -> Macro -> run the resize\_rebright (adjust the min max values and ratio according to your needs). This will downscale images to 22% of their initial size. These downscaled images are the ones that will be used for training the model and getting predictions. I use 0.22 but you can adjust it according to your needs. I used 0.22 because that would give me objects of ~25 pixels diameter, which I wanted to have. Preferably use interpolation None to cover discrepancies between conversion of images of different size.

1. For montage purposes only, run the resize\_rebright\_for\_montage\_macro. Use as input the raw images with your isolated channel (the output of step 1a). Do the same as in step 1b (except for using another macro) and save your equally sized images in a folder. NEVER use this folder for anything else apart from montage viewing because the resolution is fixed for all images!

Important : Note that for the rest of the analysis (step 6 and afterwards), you will use only the images that you got from the output of step 1b.

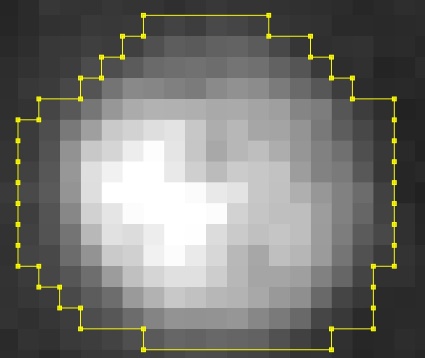
1. *Make montage of pictures from channel of interest based on the output of step 2.*
2. *Categorize images in montage based on their quality in an excel file.*
3. *Select similar ratios for training (i.e if 10 out of 100 were bad, use 2 bad images out of 20 for training).*
4. Cut squares of same size from the selected images for training (avoid having incomplete neurons at the edge of the squares).
   1. Here, you should use the output of step 1b as input. These are resized images based on a scale factor.
   2. Assess the diameter of your objects of interest to decide the size of the cropping window that you want. From Fiji *Set Scale* and then 1 pixel per unit. *Set measurements* and tick the box of area. Use a line to assess the diameter of your object of interest and press Ctrl+m to obtain the measurement. (If you want, you can have a look at the train images of the dsb2018 file of stardist and use a scale factor that produces output images that resemble the size of objects there. Objects in dsb2018 training were around 80-280 pixels).
   3. For objects of ~25 pixels, I selected 160x160 cropping window(the example given in stardist is 516x516 for 80 pixel objects).
   4. To maintain a selection of same size use the square\_selection.js or re-create it by going to plugins -> new -> javascript and typing the commands (without the green) as below. Then go to Plugins -> Macros -> Run and choose the square\_selection.js :

Move the square with the keyboard arrows for fine adjustment and once placed properly press Ctl+Shift+X or, otherwise Image -> Crop. And then save as tiff to another folder (*Folder C*).

Tip : You can run the macro with a shortcut : Plugins -> Shortcuts -> Add Shortcut -> Run… and bind it with a key. It’s smart to place a copy of the macro inside the directory of the images that you open. Otherwise you’ll have to alternate between directories for image selection and macro selection.

7) Open Qupath (0.4.3), create a folder to save your project and add your images.

8) Segment cells using Qupath (0.4.3). I personally used the brush tool because it helps to create contours similar to the shape of the fluorescent object. An example of capturing was this :



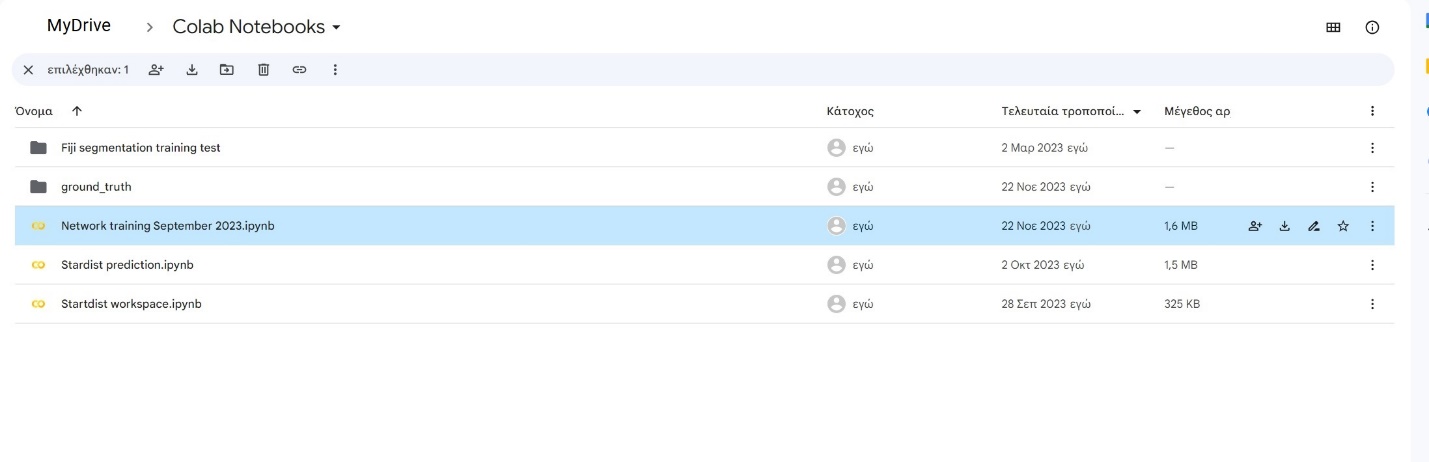
Mind the brightness levels (Tab View -> Brightness/Contrast). You might have to adjust to ‘auto‘ if your images have different noise and brightness levels. This will help you see more clearly the difference between your object of interest and your background. Keep in mind that this can affect the annotation of your contours. A darker image will result in seeing the contours closer to the center of the object. Try to maintain some consistency across images if possible. A darker background can also result of course in not labelling objects that are now too dark to spot. But you can deliberately skip this objects (that are close to the background noise) and not annotate them.

9) Create masks based on annotations. After all images are labelled, go to Workflow -> Create Script. Delete any former rows and copy/paste the entire text from ‘script\_for\_qpath’ (from the folder with the scripts) in the script editor. Select Run for Project. A ‘ground\_truth’ folder must be created with 2 subdirectories : one including the images and one including the masks.

10)Optional In case you want to visualize your masks :If you try to open a masks file from Fiji and it’s blank, try Image->Adjust->Brightness/Contrast and click on reset. If that’s the case, repeat for all. Furthermore, some masks might have a light gray that’s barely visible on your screen. You can :

10a) Image -> Adjust -> Brightness and set minimum all way to the left and maximum all way to the right (You can make a macro for that with do Process -> Batch -> Macro)

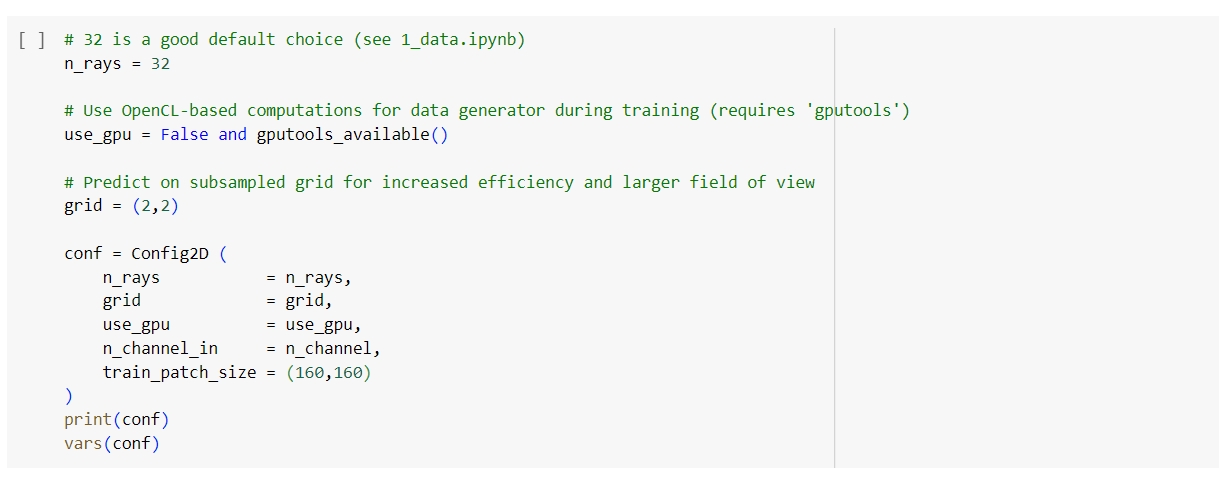
11) Skip this

12) After you make sure that the files in ‘images’ and ‘masks’ inside the ground\_truth directory have the same names, upload the ground\_truth directory in Collab and then run the script called ‘Network training September 2023.ipynb’. It should look like this :Here, there are a few things that you must change in the code, namely :

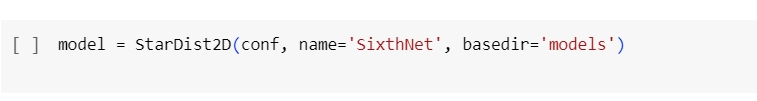
1. Depending whether your images extension is .tif or .tiff, you adjust this accordingly :



1. The train patch size should be the size of your cropping windows (provided that they all had the same size) :



1. The name of your model should be based on your input :

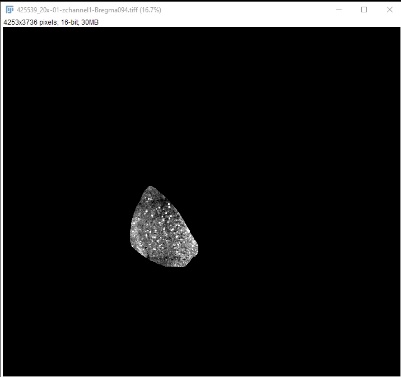


Make sure that you run all the cells one by one first and then wait for them to be executed automatically. The training of the network takes some time and if you haven’t pressed the “play” button for all the cells following the cell of training, google collab might think you are idle. Then you’ll have to do everything from scratch…

After training is finished and all the rest cells are executed, in your ground\_truth folder, a subdirectory named models should have been created. Inside models, you should be able to find your last network. Transfer it to your desktop.

13) Change background color to white and get the Mean Gray Value and Area of your segmented part only. The white background will help filtering out False Positives as well as locating the segmented part as ROI in the broader picture. The segmented part is practically everything that stands out from the background in the photo below. The mean gray value of this region will be saved and later used for visualization of each images as well as a parameter for the mean gray filter in RemoveRois.py.

* If (**and only if**) you are now introducing for the first time full-sized ‘segmented’ images with the cropped areas like this:



From Fiji : Process -> Batch -> Macro and Open the ‘resize\_rebright\_macro’. Be careful here : if you have already resized the images and you run the macro, then if you run this script you will resize them again, thereby making it impossible for the stardist network to work properly. The segmented images as given by UseROIsToSegment.ipynb, are practically raw images in terms of size,resolution and min/max display. As always, you want to use downscaled images based on the same ratio as the ratio used for the conversion of images at the first place (output of 1b). In our case, the ratio for conversion inside the macros is 0.22. Use as input the directory of your brain region of interest (which will be a subdirectory of UseROIsToSegment.ipynb output) that contains such images -of your channel of interest- and create an empty folder which you will choose as output.

* From now on you have two options :

Option 1 : Change the background color to white only (the background will then take by default the max value that you’ve set for your images in the resize\_rebright macro. For instance, 1516 in my case for cFos.

Pros : Anything that exceeds this value is taken out from the measurements, thereby the

very bright spots or cells of the image are not taken into account when calculating the mean Gray Value that we want to obtain as a background noise measure.

Cons :

If there are too many bright spots in the image, this means that much of the area is taken out from the measurements because the pixels above 1516 won’t be taken into account.

If there are too many bright neurons in the image, their mode value will surpass the mode value of the background, thereby rendering many neurons unlabeled after the filtering. Which practically means that after step 19, you’ll have to add more neurons manually.

1. From Fiji : Process -> Batch -> Macro and Open the dynamean.ijm. In the output field, chose an empty folder where the images with white background will be saved. {We need images with white background before we start predictions. The reason is that our segmented images will yield False Positives across the contours of the cropped area and if the background is white then **all** the False Positive will fall under mostly in outside area. Which means that the mode of these false predictions will be the same of the background, thereby easily removable with removeRois.}
2. Replace the output path in the dynamean macro “ *(saveAs("Results", "C:/Users/angdid/Desktop/areaout/" + title + ".csv");* “ with your path’s desired output for the measurement of the area. Replace the 1515 in the “ *setThreshold(0, 1515, "raw");* “ with your maximum value – 1. In our case, we were using maximum of 1516 for resize in the ‘resize\_rebright\_macro’, thereby we use 1515 to select everything lower than maximum (which is practically everything[or almost everything] besides the background that surrounds the cropped image).
3. Test: Run a measurement at some random background location in a few images of your output directory and make sure that the mean value that you get is the max value that you used in your pipeline (in our case 1516).

Option 2 : Change the background color to white in order to obtain the maximum display value that you’ve set in resize\_rebright macro (1516) and then convert all pixels with this value to the real maximum value for 16 bit, that is 65535.

Pros: The area will be measured more appropriately since all values below 65535 will be

Included in the measurement. Therefore, even very bright spots will be taken into

account.

Cons : The calculation of the Mean Gray Value will be done also based on the bright regions. If the very bright regions occupy much space in the segmented region, they might inflate the number of the Mean Gray Value.

1. From Fiji : Process -> Batch -> Macro and Open the 65535dynamean.ijm. In the output field, chose an empty folder where the images with white background will be saved.
2. Replace the output path in the 65535dynamean macro “ *(saveAs("Results", "C:/Users/angdid/Desktop/areaout/" + title + ".csv");* “ with your path’s desired output for the measurement of the area.
3. Replace the 1516 in the “ *changeValues(1516,1516,65535)* “ with your maximum display value that you set in your resize\_rebright\_macro.
4. Set the threshold to the real maximum value for you (number here)bit images – 1. That is, 65535-1 = 65534 for 16bit images. This will include everything apart from the background.

Remark for options :

In the future, a new macro that will combine dynamean and 65535dynamean will be made. This will combine measurements for optimal results. That is, the Area will be measured from one code and the mean gray from another. The result will be a single csv file for each image. However, for now :

For the Mean Gray value, which is the first thing that we want to obtain, we can run option 1. Remember, when the time comes to calculate area (step 22), it is **imperative** that we run option 2 too, because the area measurement might be off for some very bright images. It might also be useful to run option 2 before the filtering (step 17), because too bright neurons won’t be filtered out by inserting a mode value based on the maximum display value (1516 in my case).

14) Run the predictions based on the network.

* Transfer your newly created images with the white background to a new folder (eg. folder imagesset or another name)
* Transfer this folder to a workspace directory (e.g Inputs and Results or another name).
* Run StardistPredictions.ipynb.
* In the code below :

os.chdir(r'C:\Users\angdid\Desktop\Inputs and Results')

X = sorted(glob('BLA images/\*.tif'))

1. Replace the directory r'C:\Users\angdid\Desktop\Inputs and Results' with the directory of your workspace folder. Repeat for all identical paths inside the code of StardistPredictions.ipynb .

Replace the BLA images with the name of the folder that contains the white background images.

os.chdir(r'C:\Users\angdid\Desktop\Inputs and Results\imagesset')

1. Replace the directory r'C:\Users\angdid\Desktop\Inputs and Results\imagesset with the path that leads to your folder with white background images. Remember that the folder containing the white background images (imagesset in my case) must inside your workspace folder that you defined in step a) (Inputs and Results in my case).

if True:

    fig, ax = plt.subplots(3,3, figsize=(64,64))

c) Replace the 3,3 with a combination of numbers the product of which is equal or slightly larger

to the number of your input images. Practically, the 3,3 is the number or rows and columns.

    model = StarDist2D(None, name='SecondNet', basedir='models')

d) Replace the ‘test’ with the name of your model. Your model should be inside a folder called ‘models’ which, in turn should be a folder in your desktop.

fig, ax = plt.subplots(3,3, figsize=(64,64))

e) Replace the 3,3 with a combination of numbers the product of which is equal or slightly larger to the number of your input images.

1. Note that the extension of the images is .tif and not .tiff in my case. If you have .tiff files, you have to change .tif to .tiff everywhere in the code of StardistPredictions.ipynb.
2. Note that during the visualization of images, I have some vmin and vmax settings :

a.imshow(x,cmap='gray', vmin = 180, vmax=1100)

You can either remove them completely or change them if you want. This will not affect the final output.

The directory that will contain the output of the predictions (previous images, new zip file with rois and new images with masked rois) decided on the last path in the code :

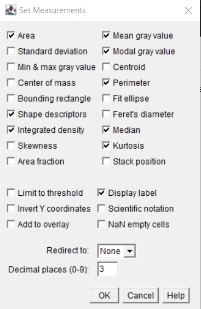
os.chdir(r'C:\Users\angdid\Desktop\Inputs and Results')

You should have already replaced that one with your path.

Note: For one set of images only (LA), I encountered an overflow error which was solved when I changed the background back to black. The only spotted difference between the LA and the rest of the datasets (bla,cea) was that there smaller area compared to the white background. It is likely that somehow this difference caused the overflow.

15. Get the all the measurements you are interested in for all images.

* Ensure that you have all measurements of interest. It is necessary to select mean gray value as measurement. Also ‘mode’ must be selected. It is highly recommends to select Area as well as Round too (which is a byproduct of other selections). From Fiji -> Analyze -> Set Measurements to add your measurements. An example of mine is below :



* Copy and paste all the resized images with white background (output of step 13) to a new folder A
* From the output of StardistPredictions.ipynb , copy and paste all .zip files that contain the ROIs to a new folder B
* Create a new folder C where the results of measurements will be saved.
* The output type must be TIFF
* Now we’re going to give the input and output paths to our macro:
  + Open the ResultsRois.ijm with a text editor and replace the path in line 5

with your path. Example : *("Open", "C:/Users/Me/Desktop/B/" + title + ".zip");* Note that we type B because this is the folder that contains the ROIs.

* + In the same file, replace the path in line 8 with your path. Example :

*saveAs("Results", "C:/Users/Me/Desktop/C/" + title + ".csv");*

Note that we type C because this is the folder where the csv measurement

results will be saved.

* Open Fiji and run Process-> Batch-> Macro and select ResultsRois.ijm. (Keep in mind that by default the last used macro will be written there. Therefore make sure you open the last version of ResultsRois.ijm that you have every time).
* Set folder A as input and delete anything that’s written in the output field of the interface.
* Process.
* There is a chance that you will get an error if files cannot be traced properly. Check your path carefully to see why your files can’t be traced. Lastly, if you are using Mac instead of windows, it is likely that you’ll have to change the format of these paths.

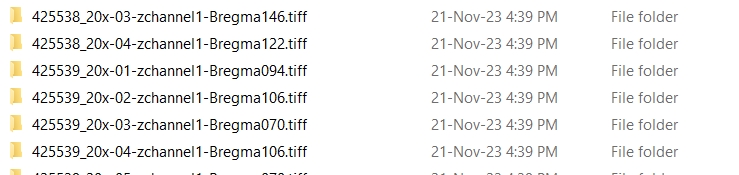
16. Rename the ROIs from 001\_001, 001\_002, 001\_003 to 1,2,3 etc. respectively.

* From the output of StardistPredictions.ipynb , copy and paste all .zip files that contain the ROIs to a new folder A. Unzip files to separate folders and delete the zip files. This is your main ROI directory. Run the rename rois.py. First select this folder A that you made and then select an empty folder B.

17. Filter the ROIs based on cut-off criteria. Run the RemoveRois.py and insert the cut-off criteria of your choice. If a ROI is excluded in at least 1 criterion, then it will be excluded. For the time, ignore the ‘Double’ option in the area. This might be a future option.

* Mind that for then Mean Gray Value and only, the value which you are inserting is **not** an absolute value but an additional value of your choice which is added to the value of the mean background of the segmented area of each brain region (this background was measured in step 13). For the time, only provide input to the **min** field of the Mean Gray Value and not to the max field of Mean Gray value. This way, you are practically saying that if the mean background of that region + the value that I insert is at least n, then you accept the ROI which was predicted by stardist. If you leave the min field of Mean Gray empty, you are practically setting as minimum value the value of the background of the respective image across the iteration. You can define both min and max for the rest of the filters in the GUI.
* For the Mode field, if you are using the regular snippet, you should be able to see two entry fields, one for minimum value and one for maximum. Set 0 and 1515 (that is, background pixel value -1) for minimum and maximum,respectively.
  + If you are using the version with one entry field (inside the obsolete folder) type the pixel value of your white background in your white background images (it should be the same). This way, the False Positives around the contours will be excluded. This second script hasn’t been tested thoroughly and it might yield unexpected results.

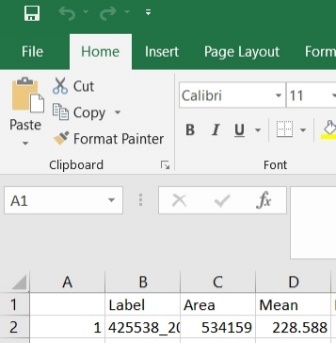
Select the roi folder, which should be a directory with subdirectories that contain the **renamed** roi files. It should look like this :



Select the nuclei measurements which should be a directory with the csv files that contain the measurements that you’ve run in step 15.



Select the folder with the background measurements (these are the mean gray value measurements that you took in step 13). Mind that it might be preferable to use background measurements based on option 1 of step 13 for that. The names of the files should be the same as the csv files above, however, inside they should look like this :



Select an empty output folder.

When you’ll press ready, if the code runs well, an ‘excluded’ and ‘final\_rois’ folder will be created in the output folder. Inside them you’ll find the ROIs which are included an excluded for each animal following the filtering. (**Note that if you press ready, the two subdirectories will be created inside this folder even if the code yields an error. In this case you have to delete these subdirectories until the code runs properly**).

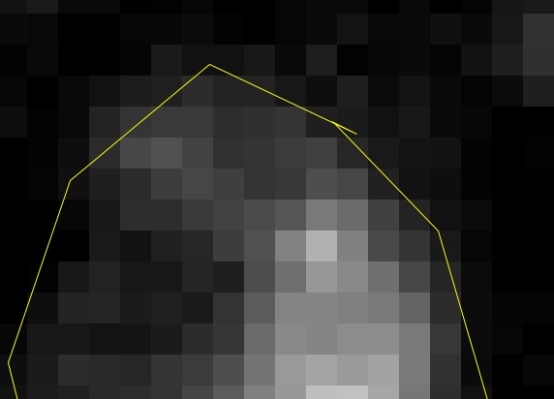
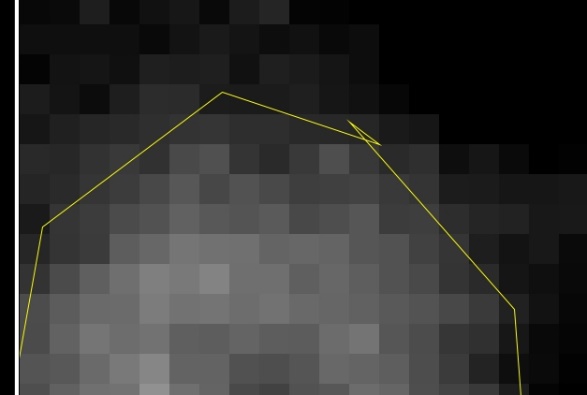
Optional : The script includes instructions in case you want to add further measurements to the GUI. As a rule of thumb, you can search all instances of a label and field variable and create a new label and field variable accordingly. Mind the global variables too, as well as the max Variables and the fact that they are named as the min variables with a max in front of the name. Also note whether there are measurements in the csv files the name of which ends with a dot. In this case you can either create a script to rename the column (in all csv files) in order to get rid of the dot or fix the problem inside the code instead, as it’s done for the Perim. (perimeter). The second option is probably more complex than the first.

18. Change min and max display before salvaging the erroneously excluded rois. We will use the mean gray value of the whole region as minimum value. Minimum value \* 2.7 will be the maximum value of display. Process -> Batch-> Macro -> Open ‘meanforminimum.ijm’

* in the line : open("C:/Users/angdid/Desktop/background/" + title + ".csv"); replace this path with the output path with the mean gray measurement of each region (that is, the background measurements which is the csv output of step 13).
* In the line : maximum\_display = mean \* 2.7 replace the 2.7 with a parameter that optimizes the display of your images. In my case, using the maximum value of display was optimal with values around 2.7 – 2.9.
* In the line : savepath = "C:/Users/angdid/Desktop/newminimum/" replace this path with the path of your own directory to save the output images.

19) Correct manually by a) adding false negatives b) removing false positives c) restoring false negatives

Disclaimer : If you are planning to use the colocalization snippet too, it is important to annotate new objects with freehand selection (or polygonal selection). Y and X coordinates are not submitted when annotated using circles. As a result, the coordinates of these regions will not be properly submitted and analyzed. Also avoid drawing peculiar contours in neurons that you want to add. For instance, below are two cases that won’t be analyzed by the colocalization script (and will be submitted as warnings in the log file)



* From Fiji -> Process -> Batch -> Macro and open manual.ijm (if you are using blinded images and zip files based on blind.py see the note under these bulletins)
* Use the output of step 18 (the savepath) as input. These are the images with the adjusted minimum and maximum display.
* Delete anything in the output field
* In the line : excluded\_path = "C:/Users/angdid/Desktop/filtered/Excluded/" replace this path with the path of your excluded rois (output of step 17)
* In the line included\_path = "C:/Users/angdid/Desktop/filtered/Final\_ROIs/" replace this path with the path of your filtered-in rois (output of step 17)
* Mind that the excluded and included path must contain both the zipped and unzipped versions for the subfolders containing the rois.(This means that in the case of using blinded images based on step 19a, then you must unzip the renamed zip files too)
* In the line roidir = "C:/Users/angdid/Desktop/Final Results/" replace this path with the path that you want to save your final results
* Run the macro. Cyan ROIs will be the filtered-in. Yellow ROIs will be the filtered out.
  + If any neurons are not spotted by stardist (and you are done with refining the network or lowering the probability threshold to include more detections) you can draw a ROI using Fiji and press *t* to add it to the ROIs.
  + If any cyan ROIs do not represent real neurons/nuclei, you can remove them by pressing on their number and then *Delete*
  + If any yellow ROIs include real neurons, you can salvage this erroneously filtered-out neurons by just deleting the ROI that surrounds them.
* Keep this line of actions to make sure you never re-draw manually a ROI that you previously salvaged by deleting the yellow ROI of an erroneously filtered-out neuron.
* ENSURE that you keep a backup after you’re done. These data are your manual corrections and cannot be retrieved!

Notes : If you are using this macro after blinding your images and zip files first (as described in 19a), then run this macro instead : manual\_for\_blinded.txt

You can also go to plugins -> macro -> install and select the show all.txt. This will allow you to show all rois with d and show none with a. You can remove a selected roi with s. You have to install the macro every time you open fiji. Otherwise you can add it to the autostartup inside fiji directory. Mind that this macro has not been tested in combination with the macro of step 18, therefore run some tests to see if the ROIs are saved according to your actions first.

19a ) Optional : Blind yourself by watching Pokémon :

* Create a backup of your filtered-in and out ROIs (output of step 17) as well as your adjusted for minimum and maximum display images to ensure no loss of data(output of step 18). Preferably, also create copies and work with them, instead of the initial files.
* Create a folder where the renamed tiff images will be saved
* Run blind.py
* The blind.py works for maximum 200 files but you can enlarge the list inside. ChatGPT can produce long lists with names of whatever you want. You can then paste this list in the code and use it for blinding.
* Select the directory with your images first
* Then select the output directory. This is the directory where your newly produced renamed images will be made
* Then ensure that you have a copy of your ROIs and selected one of the two directories (either Excluded or Final\_Rois)
* Repeat the same procedure for the other.
* WARNING : If you accidentally make a mistake and have to rerun the program, ensure that you delete the newly produced renamed images in the output directory. These images are produced before the program finishes. There are safety valves, so you cannot practically re-use the same output directory unless you empty it first. But in any case, empty it yourself after failed attempt to run the program properly.
* WARNING : Do not delete the ‘encrypted’ file inside the folder with the renamed images.

Unblind yourself

Run the decode blinding.py (for ending with .zip or .tiff zip depending on the format of zip files you have) to unblind the images (if you’d like) and the decode blinding for zipped to unblind the zip files.

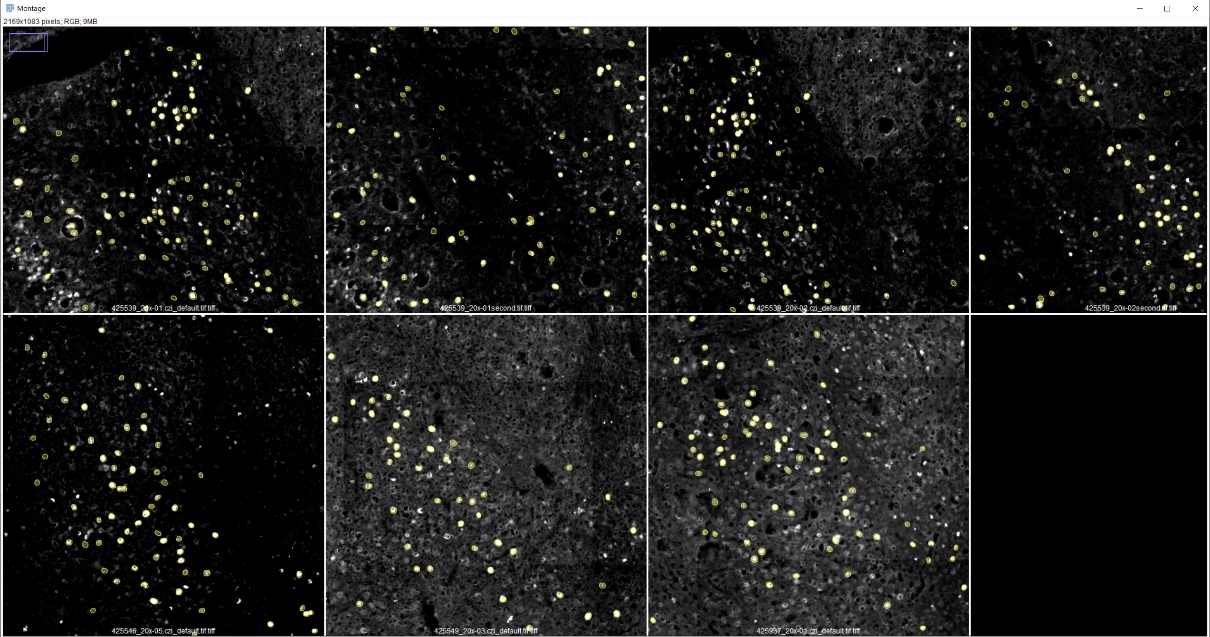
* Assuming that you open the decode blinding for zipped, first locate the file called ‘encrypted’ inside your folder with the renamed/blinded images.
* Locate your file with zip folders.
* Select a save folder

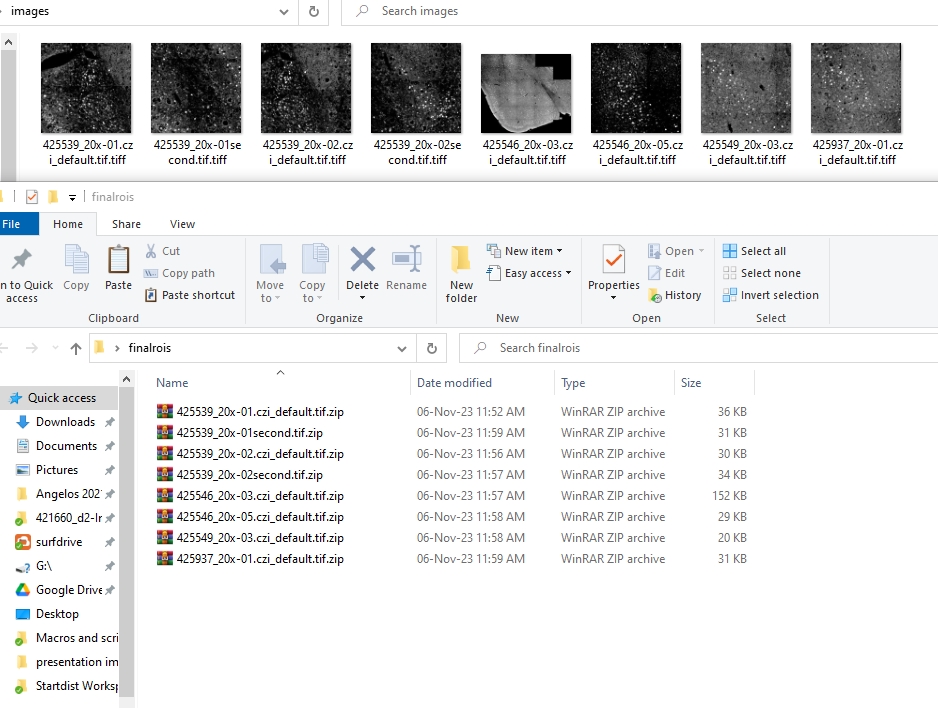
Get-ChildItem -Filter "\*.zip\*" | Rename-Item -NewName { $\_.Name -replace '.zip', '.tiff.zip' }

20). Visualize the ROIs in multiple images.

Option 1:

We will make again a montage, this time containing the final predictions. Will look like this (note that this is not in fact my final montage. Only for presentational purposes) :



* First, we need to create .zip files with all .roi files inside for each image. This is because imageJ reads .zip files for ROIs. Here’s a screenshot with the names of the images files on top and the names of the created zip files at the bottom. To create a .zip file, simply select all .roi files inside the Final\_Rois folder (created in step ..), right click -> send to -> compress to zip. You can also make a script on your own to avoid manual conversion of rois foilder to zip. 
* Create a folder (e.g images) and insert the images with the new minimum and maximum value that you produced at step 18.
* Create an empty folder (e.g finalRGB)
* Then open the finalmontage.ijm and
  + Replace this path -> *roiManager("Open", "C:/Users/angdid/Desktop/finalrois/" + roititle + ".zip");* with your path. This is practically the path you see in the finalrois picture above.
  + Replace this path -> saveAs("Tiff", "C:/Users/angdid/Desktop/finalRGB/" + title); with your path to the empty folder.
* Process->Batch-> Macro -> and open the finalmontage.ijm (make sure you open it again every time you make changes in the macro because previously opened versions appear as the default).
  + Insert the folder images as input.
  + Delete everything in the output.
  + Save as tiff.
* The output images including the ROIs must be saved in the finalRGB folder.

Option 2: The finalRGB images have less pixel range. As a result, some of your images might be too bright to inspect the predicted ROIs. You can then go image per image then. Run validate.txt and replace

*roiManager("Open", "C:/Users/angdid/Desktop/tests/ziped finalend corrected/" + title + ".zip");* with your path that contains your ROIs.

21) Get the final nuclei measurements of your manually corrected ROIS. Repeat step 15, yet with your Final ROIs (output of step 19).

* Ensure that you have all measurements of interest. It is necessary to select mean gray value and area as measurement. From Fiji -> Analyze -> Set Measurements to add your measurements
* Copy and paste all the resized images WITHOUT the white background, that is, the ones you used as input in step13 (the white background images will affect some measurements so you don’t want them) to a new folder A.
* From the output of step 19, copy and paste all .zip files that contain the ROIs to a new folder B.
* Transfer zip files to a separate folder (make sure you have a copy of them before you proceed).
* Create a new folder C where the results of measurements will be saved.
* The output type must be TIFF
* Now we’re going to give the input and output paths to our macro:
  + Open the ResultsRois.ijm with a text editor and replace the path in line 5

with your path. Example : *("Open", "C:/Users/Me/Desktop/B/" + title + ".zip");* Note that we type B because this is the folder that contains the ROIs.

* + In the same file, replace the path in line 8 with your path. Example :

*saveAs("Results", "C:/Users/Me/Desktop/C/" + title + ".csv");*

Note that we type C because this is the folder where the csv measurement

results will be saved.

* Open Fiji and run Process-> Batch-> Macro and select ResultsRois.ijm. (Keep in mind that by default the last used macro will be written there. Therefore make sure you open the last version of ResultsRois.ijm that you have every time).
* Set folder A as input and delete anything that’s written in the output field of the interface.
* Process.

22) Get the number of neurons, area size and average intensity for each image.

* Ensure that you have a background measurements **based on option 2** of step 13.
* Warning For Future Code : If a function for intensity will be added in the future, ensure that you have nuclei measurements which are taken based on the last ROIs of step 19. If you go completely automatic and you skip manual rectification (step 19) then ensure that you have the primarily resized images before the white background. If the measurements are taken from an output of option 2, then the neurons that will have pixels with the very specific maximum display value (1516 in this case) will skyrocket to 65535. This will change the measurement of their mean gray value dramatically.
* Open the Neurons\_per\_Area.py \*
* You will be prompted to select the right directories for

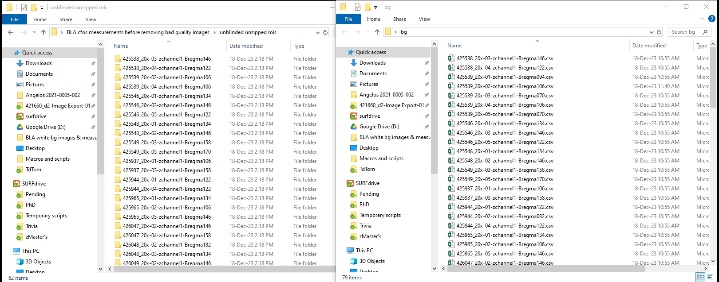
1. The folder containing the unzipped ROIs into separate folders. Example :



1. The folder containing the background measurement (practically the area) of each segmented region. These are .tiff.csv files. So practically, same as in folder name with a .csv extension



Another example of input (without the .tiff extensions) is this :



Although you should probably have the folders and csv files containing the .tiff part.

1. Optional. This window will appear if you uncomment the

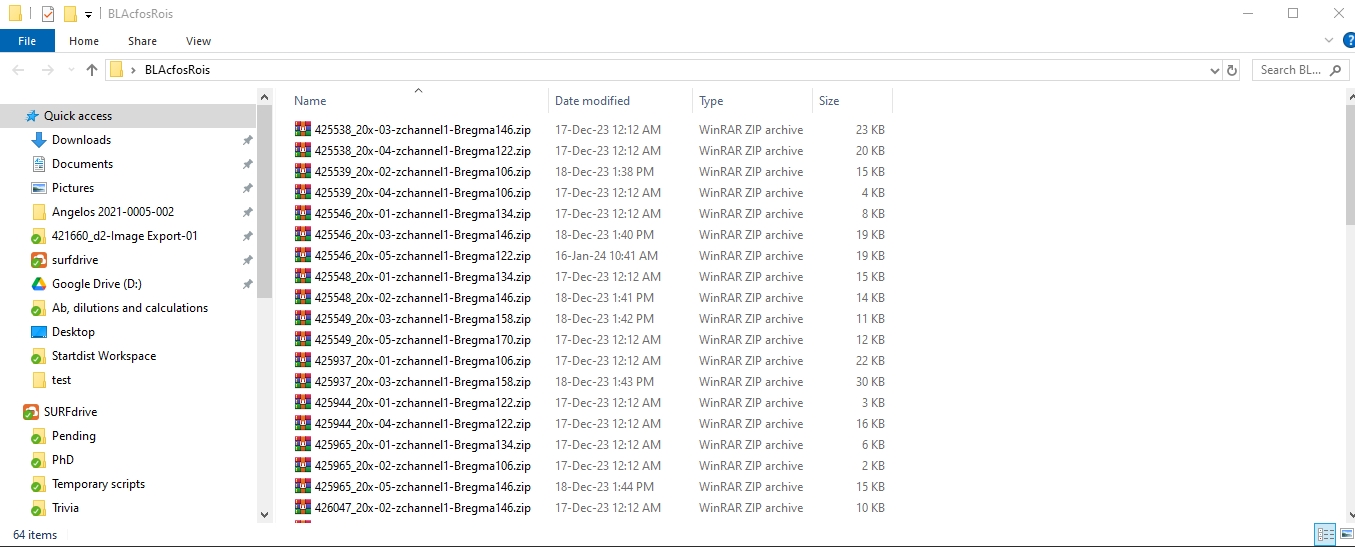
# SelectNucleiMeasurementsDirectory ()

# GetAverageIntensity(measurement\_path)

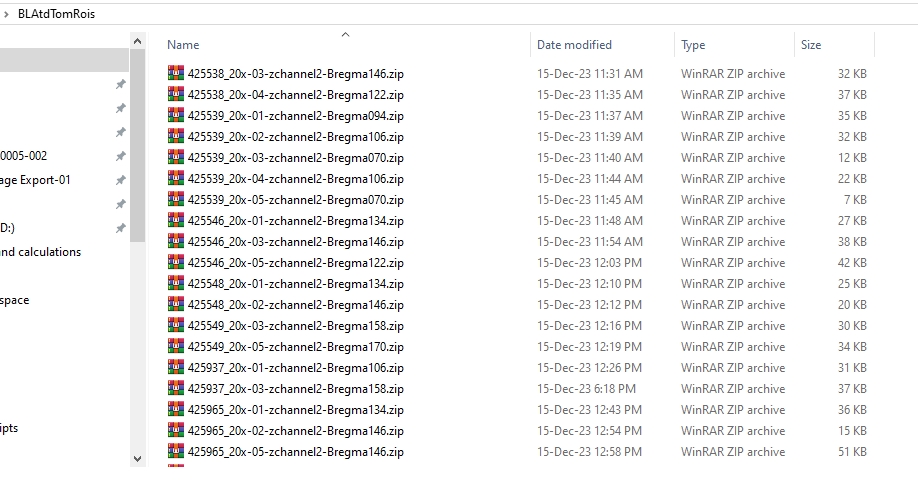
in the snippet. This is for future use and not for now. This folder must contain your **NEW** nuclei/cell measurements that you’ve taken at step 21.

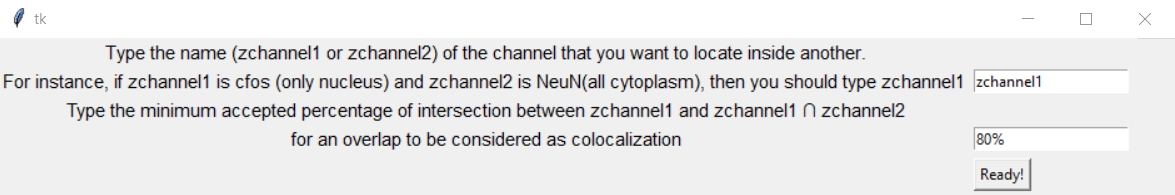
23) Colocalization of 2 channels

Run the colocalization.py

1. Select the first directory containing zip files (which have inside roi files)
2. Select the second directory with files of the same type (but from a different channel this time)

The filenames should be the same, except for the part that distinguishes zip files of zchannel 1 and zchannel 2.



1. Follow the instructions on the 3rd popup screen and type the part of the name which is different between two zip files of the same image but containing rois from a different channel. You do this for both channels. In this case it is zchannel1,zchannel2
2. Leave only the name of the channel that you want to study overlap **inside** another. In this case, zchannel 1 is cfos(nucleus) and zchannel2 is tdTomato(cytoplasm). So I leave only zchannel1.
3. Select the minimum accepted overlap for your channel with the smaller objects.
4. Select a directory to save your data.
5. A colocalization excel file with a timestamp will be created. In addition, a text file called colocalization\_log(check\_for\_WARNINGS) will also be created. Open it and see if there are any WARNINGS. If so, test the particular rois that yielded warnings visually for colocalization.

General :

If you have to convert names from .tif to tiff, you can open the powershell , cd your folder and run

Get-ChildItem -Filter \*.tif | Rename-Item -NewName { $\_.Name -replace '\.tif$','.tiff' }

If the file name is longer (i.e .tiff.csv ) and you want to cut a specific part, then instead of \*.tif | do \*.tiff.csv

\* For the Neurons\_per\_Area I used the DLC\_env for the dependencies. I can change that.

Yaml file for each script :

Remove\_Rois -> pyqt5