QDyeFinder Basic Read Me

# Requirements

* [Bioformats MATLAB toolbox](https://www.openmicroscopy.org/bio-formats/downloads/)
* MATLAB (tested on v9.11)

NB this is designed to be a stand alone version – any non-published extra code written by me (Marcus Leiwe) should not be needed to be downloaded, with the exception of the [non-linear correction](https://github.com/mleiwe/HyD_NonLinearCorrection) and [linear unmixing](https://github.com/mleiwe/LinearUnmixing) repositories which may or may not be needed depending on your imaging set up.

# Download and Installation

Simply download this repository and set the path (with subfolders) in MATLAB. This readme guide will guide you through the main steps of QDyeFinder

# Example Data

Due to image size, we have provided a small image that can be used to run the pipeline with the .xml files from Neurolucida360 also available. The outputs are also available inside the repository.

For additional information on QDyeFinder please see our paper on [biorxiv](https://www.biorxiv.org/content/10.1101/2022.10.20.512984v1)

# Process Image

## Standard way

Assumes that [non-linear correction](https://github.com/mleiwe/HyD_NonLinearCorrection) and [linear unmixing](https://github.com/mleiwe/LinearUnmixing) has been performed. Therefore this portion performs: the chromatic aberration correction, loading and saving into MATLAB as variables, and creating a greyscale for tracing in Neurolucida.

Our small size demo image and workspaces are available on the QDyeFinder repository.

1. Type in [cData,Scale,dim]=mnl\_ProcessImageForNeurolucidaImages
2. You will then be prompted to open the image we want to process.
3. The next stage is to correct the chromatic aberrations. This is based on the machine learning solution published [here](https://www.frontiersin.org/articles/10.3389/fnana.2021.760063/full). If ‘y’ is entered then the chromatic aberration correction is performed requiring the following steps.
   1. You will be asked to enter the excitation laser wavelength for each channel
   2. Then you will be asked which channel you want to correct to (i.e. which channel is the reference channel)
   3. Finally you will be prompted to load in the workspace containing the linear regressions as published in [Leiwe et al 2021.](https://www.frontiersin.org/articles/10.3389/fnana.2021.760063/full)
      1. An example is provided within the QDyeFinder repository inside the processed traces folder entitled ‘ChromaticCorrections\_SeeDB\_x63.mat’
      2. NB This is bespoke for each microscope, and if the laser paths have been re-aligned, etc.. this may change. If you are using your own image do not use the correction file provided as a demo
4. Then the final stage is when the program creates a grayscale used for the tracing programs such as Neurolucida 360.
5. This workspace is saved as “ProcessedImage.mat” with the variables described in the docstring.
   1. If you have already performed the tracing and image processing somewhere else feel free to load your image as the variable “cData” with the xyz scale as “Scale”, and the xycz dimensions as “dim”.

# Process Traces/Somas

Currently QDyeFinder is only functional with Neurolucida360 files, that have been saved in the .xml file format. A demo data set will be available in the repository section as well. (Somas.xml = Somas, and Traces.xml=Neurite traces), this must be used in conjunction with the data files available here (link to RIKEN repository) which will be available soon.

## Process Somas

1. Type [Somas,BkgMean,BkgStd,Thresh]=mnl\_ProcessTracesFromNeurolucida\_Somas(fname,cData,Scale,dim)

Where ‘fname’ is the filename (without the extension) of the location of the somas. And cData, Scale, and dim are the outputs from mnl\_ProcessImageForNeurolucidaImages. The following information describes the user prompts in more detail

* 1. Do you want to manually select the background? (y/n)
* If ‘y’ is selected then you will be asked to draw a rectangle over a MIP of the image which will generate a cuboid where all the voxels inside will be considered as the background voxels
* If ‘n’ is selected then the background voxels will be a randomly chosen subsample of all voxels not chosen within the Neurolucida file.
* ‘n’ is recommended for processing time.
* Scatterplots of the background and the selected voxel intensity that compare the values between each channel is provided as a pop up figure that can be used to evaluate whether there is a correlation between channels (usually this means something has gone wrong in your imaging).
  1. Do you want to filter the channels manually (y/n)?
* If ‘y’ is selected then you will be shown a cumulative plot comparing the “Signal” and “Background” voxels along with the signal to noise ratio at each percentile. It is then dependent on the user to decide whether there is sufficient signal in each channel
* If ‘n’ is selected then the mean signal to noise ratio between the 80th to 100th percentile must be above the signal threshold value (typically =3)
* ‘y’ is recommended if you want to see the contribution of each channel more clearly, but it is not necessary

1. The output file is automatically saved as ‘ProcessedSomas.mat’ with the outputs described in the docstring.

## Process Traces

Type [efPxTrace,Somas,BkgSigCumulativeDistribution,NumLengthThresh,MagThresh,BkgMedian,MaxValues]=mnl\_ProcessTracesFromNeurolucida\_DendritesSomas\_v14(fname,cData,Scale,dim)

1. Where ‘fname’ is the filename (without the extension) of the location of the somas. And cData, Scale, and dim are the outputs from mnl\_ProcessImageForNeurolucidaImages. The following information describes the user prompts in more detail
   1. Do you want to correct the automatic Neurolucida tracing?(y/n)

Occaisonally, the Neurolucida algorithms select the edge of a neurite rather than the middle. If you select ‘y’ this option will attempt to correct this by finding the location of the true middle.

* 1. Do you want to manually select the background? (y/n)
* If ‘y’ is selected then you will be asked to draw a rectangle over a MIP of the image which will generate a cuboid where all the voxels inside will be considered as the background voxels
* If ‘n’ is selected then the background voxels will be a randomly chosen subsample of all voxels not chosen within the Neurolucida file.
* ‘n’ is recommended for processing time.
* Scatterplots of the background and the selected voxel intensity that compare the values between each channel is provided as a pop up figure that can be used to evaluate whether there is a correlation between channels (usually this means something has gone wrong in your imaging).
  1. Do you want to filter the channels manually (y/n)?
* If ‘y’ is selected then you will be shown a cumulative plot comparing the “Signal” and “Background” voxels along with the signal to noise ratio at each percentile. It is then dependent on the user to decide whether there is sufficient signal in each channel
* If ‘n’ is selected then the mean signal to noise ratio between the 80th to 100th percentile must be above the signal threshold value (typically =3)
* ‘y’ is recommended if you want to see the contribution of each channel more clearly, but it is not necessary
  1. Do you want to calculate the minimum fragment length for colour consistency? (y/n)
* If ‘y’ is selected, then an algorithm is run which calculates the minimum distance of a fragment where the colour is of a consistent value
* If ‘n’ is selected you must then enter a minimum length of a trace.
* If you are familiar with or data, or have already run similar images then it will recommend to select ‘n’ to save time. In our images we found a minimum distance of 5um (for x63 images) or 10um (for x20 images) worked best.
  1. Do you want to determine the minimum brightness of the trace? )y/n
* ‘y’ runs the algorithm to find the minimum brightness as represented by the magnitude of the normalised colour vector (NB the step prior to the vector normalised version)
* ‘n’ will require you to input a value, typically between 0-1 is recommended. In our experiments we find a value of 0.1 produces good quality information
  1. Do you want a figure of the final processed traces? (y/n)
* Fairly self-explanatory, a final figure of your traces is produced. Select no, if you want to save time and processing power.

1. The output file is automatically saved as ‘ProcessedTraces.mat’ with the outputs described in the docstring.

# Establish the threshold

Before running the dCrawler algorithm it is recommended to determine what is a good threshold would be by manually tracing values in Neurolucida, and assigning traces with that belong to the same group in the same set in Neurolucida360 (stored in the ‘SetId’ field of the efPxTrace structure in QDyeFinder). Example file provided as “ProcessedTraces\_Manual.mat” in the repository

Once that is done run

[OptF1mean,OptThresh,OptF1median,OptThreshMedian]=mnl\_OptimumEuThreshViaF1Star\_MagWeighted(efPxTrace)

Where ‘efPxTrace’ is the output from 2. Processed Traces.

This will provide the optimum threshold value (OptThreshMedian) as calculated via the F1

The F1 scores are also provided.

# Clustering

Now that the optimum threshold is determined (or you decided to skip step 3) run

[FinalClusters,FinalClusterIDs,NumTraceLim,MinDendriteLength,InitialClusterNum,Y]=mnl\_EuclideanCrawler\_Mag\_Weighted\_v3(Trace,EuThresh,dim,FigureDisplay,Y,MinPoints,MinLength)

Input and Outputs are explained in the docstring but briefly…

**Inputs**

**Trace** – This is the trace structure produced by mnl\_ProcessTracesFromNeurolucida\_DendritesSomas\_v14. Typically termed efPxTrace

**EuThresh** – The threshold for dCrawler

**dim** – The dimension of the image (this should be automatically saved by mnl\_ProcessTracesFromNeurolucida\_DendritesSomas\_v14

**FigureDisplay** – a ‘y’ or ‘n’ to indicate whether you want to visualise your clusters as maximum intensity projections (MIPs)

**Y** – tSNE location values, if they are not present please leave as empty []

**MinPoints** – The minimum number of points a cluster needs in order to be retained. Usually not used (a value of 1 is entered). But it can be applied if desired.

**MinLength** – The minimum cumulative length of fragments in the cluster (in um). Usually not needed (i.e. =0).

**Outputs**

**FinalClusters** – A structure containing the information on each cluster

**FinalClusterIDs** – A simple array where each row corresponds to each trace (in efPxTrace) and the value indicates which cluster the trace belongs to.

**NumTraceLim –** The Minimum number of points for a cluster to remain

**MinDendriteLength** – The minimum total length of neurites for a cluster to remain

**Y** – The locations of each traces colour in tSNE space.

# Additional questions?

Need any more help? Please contact the lab via DM on twitter @TakeshiImaiLab. Or use the correspondence address on our biorxiv paper. (biorxiv.org/content/10.1101/2022.10.20.512984v1.full)