2023 update

qPAINT Analysis Pipeline Guideline

This guideline should help you through processing DNA-PAINT images to get quantitative information out. If you have any questions please contact me at ucbtmdj@ucl.ac.uk . It is usable for one colour or multi colour data.

Prestep: selecting ROIs from images

Overview: Before running any qPAINT analysis you need to decide whether you will analyse whole cells or ROIs from your cells. If you will analyse ROIs you first need to select these. This section will walk you through how to do this.

Required software:

- MATLAB (license required)
  - https://uk.mathworks.com/products/matlab.html
- RegionFinder
  - https://github.com/quokka79/RegionFinder

Method:

1. Convert HDF5 files (of ROIs or cells) to txt files if need be by running the MATLAB code titled **'Converting_hdf5_to_txt.m'**
   i. This code will save the x, y, frame, photon, sx (error in x) and sy (error in y) columns from the HDF5 files in that order.
2. Open RegionFinder in matlab by going to apps and clicking on the RegionFinder app
3. Open your txt cell file in RegionFinder by selecting XY only, image size of around 60 (you may need to play with this), 1nm/px (we already converted them), and a data delimiter of comma
4. Select the columns of your x,y, and frame data which should be x=1, y=2, channel ID = 0 and frame = 3
5. Select the size of the ROI in region size e.g. 3000 is 3 micrometers by 3 micrometers
6. Make sure hide first is unselected
7. Then click open one file and select the file to load
8. Then click new region, a square should appear, click and drag the square to where you want your ROI and double click to select the region
9. Once you have finished selecting the ROIs save them by selecting export regions
10. This should save a coords.txt file that you can use the cut regions of the cell in the next section of code
11. Save the file that you wish to cut into ROIs in a new folder and rename the file 1.csv
12. Save the coords.txt file also in this folder
13. Open the coords.txt file (in any text file viewer) and change the first column to be 1, change the second column to be the ROI number e.g. 1 for the first row, 2 for the second, and so on
14. Open the code file **'ROI_generation_via_coord_2023'**
15. Run this code and select the coords.txt file
16. Once this has run there should be a separate folder for each ROI with the name of the ROI in it.

2023 update

Step 1: single molecule localisation clustering using DBSCAN

Overview:
This will cluster the single molecule localisations from the DNA-PAINT image into clusters of various sizes that we will then use the qPAINT pipeline on to determine how many proteins are in each of these clusters. There are various clustering methods to do this but right now we use DBSCAN. You need to input two parameters: minpts and eps. Minpts = the minimum number of points to be classed as a cluster, this depends on the imager you use and the number of frames you take and is typically around 10, eps = the radius of the circle in which points will be considered in a cluster which is determined by the localisation precision of your images again, this is typically around 10 for ours. Running DBSCAN on a file will generate two extra columns of data for that file, the first is the DBSCAN_ID which is the cluster number that the point belongs to and the second is the DBSCAN_TYPE which is 1 for core cluster points, 0 for edge cluster points. Both parameters will be -1 for all points not in a cluster.

See more explanation here: https://en.wikipedia.org/wiki/DBSCAN

Required Software:

- MATLAB
  - https://uk.mathworks.com/academia/tah-portal/university-college-london-649021.html
- PALMsiever
  - https://github.com/PALMsiever/palm-siever/wiki
  - make sure download PALMsiever from the GitHub following the instructions to install WITHOUT DipImage

Method:

1. Convert HDF5 files (of ROIs or cells) to txt files if need be by running the MATLAB code titled **'Converting_hdf5_to_txt.m'**
   i. This code will save the x, y, frame, photon, sx (error in x) and sy (error in y) columns from the HDF5 files in that order.
2. Open PALMsiever in matlab by navigating to the file where PALMsiever is saved in your laptop and open PALMsiever in matlab by typing PALMsiever into the command line.
3. Load your txt file of the cell/ROI by clicking file, import, generic text file
4. Go to plugins, cluster_DBSCAN to start DBSCAN clustering and input the parameters for minpts and eps, this will start the clustering algorithm, you won't see anything until the algorithm is finished, then it will show you your clustered image.
5. Save the DBSCAN output to a txt file by running the MATLAB code titled **'Saving_DBSCAN_data_as_txt.m'**
   i. You will need to use the file named **'Saving_DBSCAN_data_as_txt_RegionFinder_compatible.m'** if you selected the regions via RegionFinder
   ii. This code will save the frame, x, y, photons, error in x, error in y, dbscan_ID, dbscan_type into a file with the name of the Eps and MinPts parameters in.

Extra: You can also use the code named **'drawing_DBSCAN_maps'** to draw the maps that DBSCAN computes, this is useful for figures.

Step 2: qPAINT index calibration

Overview:
This code will run the qPAINT calibration pipeline to determine the dark time for each cluster, 1/dark time is the qPAINT index. We then plot the qPAINT index of all the clusters that are smaller than 300nm in diameter in every image generated from one docking-imager strand pair. We will then fit this plot with an equation to find the qPAINT index corresponding to a cluster of 1 protein.

Required Software:

- MATLAB

Method:

1. Open the matlab code **'Internal_Calibration_02_08_2021'**
2. Input the correct user parameters into the code as requested in the code
3. Run each section of the code separately, you should get a txt file with the calibration information for that ROI at the end
4. Copy all the calibration files of a particular imager to a separate folder and call this folder concatenated files, save the files with the names ROI1, ROI2 etc. with each new file having a different number – we don't care about which cell they have come from at this point as this is just for calculating the qPAINT index
5. Navigate to this folder and run the code 'concatenating internal calibration files' to concatenate all the internal calibration files together
6. Then use the "Plotting qPAINT histogram' code to plot the histogram of all the qPAINT indexes
7. Go to apps, Curve fitting and input x as bins and y as counts to show your histogram plot
8. Click on the drop down menu for the type of fit and fit a gaussian function to the graph, see whether a two, or three peak gaussian (or more) fits the data better.
9. Look at the parameters of the fit, b1 will be the first qPAINT index (I.e. corresponding to a cluster of points for one protein)
    i. You can also check the numbers make sense by seeing if the other b parameters are multiples of the first (i.e. b2 is double b1)
    ii. You can also try fitting a custom equation using the gaussian function but replacing b2 with 2*b1 and b3 with 3*b1 (and so on) to see if this gives a more accurate number.
10. Make sure to save the graph in sfit format with file, save session and make a note of the qPAINT index for one protein – you will need this in the next code file.

Step 3: Protein quantification and mapping

Overview:

2023 update

This code will run the protein quantification which will quantify the amount of proteins in each cluster using the qPAINT index you calculated in the previous step. It will then use k-means to map the position of the proteins in a cluster onto the single molecule localisations.

For more info about how k-means works see: https://en.wikipedia.org/wiki/K-means_clustering#:~:text=k%2Dmeans%20clustering%20is%20a,a%20prototype%20of%20the%20cluster.

Required Software:

- MATLAB

Method:

1. Open the matlab code **'Protein_Map_Converter_14_04_2022_SS'**
2. Input the correct user parameters into the code as requested in the code, ensuring you input the qPAINT index you just calculated
3. Run each section of the code separately, you should get a txt file out with all the information for each cluster including the number of proteins
4. You will also see the protein maps, you can save these as matlab fig files and you will also get a csv file with the protein map coordinates that you can plot separately if need be.

Extra: You can also use the code named **'drawing_protein_maps'** to draw the maps of the proteins this is useful for figures.

Extra: You can also plot the protein maps over the DBSCAN maps which may also be useful for figures by saving the graphs separately as fig files and merging them using the code **'merging_2_fig_graphs'**

Post qPAINT analysis

Now you can use the quantitative information to compare conditions in whichever way you like! Protein numbers, cluster numbers, cluster density, percentage of proteins in clusters, cluster sizes in terms of number of proteins and cluster sizes in terms of diameter are all parameters you can look at for example. You can also do nearest neighbour distance (NND) analysis which will show you the distances of one protein to its nearest neighbour. This can give you some information about how dense the clusters are and if proteins are interacting with one another.