

qPAINTer Documentation

Philipp Steen, Feb. 2022

1. Description
2. Installation & Dependencies
3. Instructions
4. Function Overview

1. Description

qPAINT (Jungmann *et al.*, 2016) is a DNA-PAINT technique that enables integer counting of binding sites too close to each other to resolve using super-resolution microscopy. This is achieved using the calculable binding and unbinding kinetics of dye-labeled imager strands to complementary, target-bound docking strands of DNA. Given a binding site that is bound by an imager strand at a given frequency, an area of interest blinking at twice that frequency contains two binding sites.

This code performs the calculations necessary for qPAINT analysis and provides multiple consistency checks so the user may critically examine the qPAINT calculations and results.

2. Installation & Dependencies

qPAINTer consists of a single python file containing all necessary functionalities. It can be run without installation or compilation.

Requirements:

Python 3.9.x

Pandas 1.3.x

Matplotlib 3.4.x

Numpy 1.21.x

Scipy 1.7.x

qPAINTer was tested using the following versions:

Python 3.9.7

Pandas 1.3.4

Matplotlib 3.4.3

Numpy 1.21.2

Scipy 1.7.1

qPAINTer was tested on the following operating systems:

Mac OS 12.2, Windows Server 2012 R2

3. Instructions

User parameters are specified at the bottom of the file. Any number of qPAINT instances can be defined and analyzed sequentially.

To define a qPAINT instance, a number of parameters must be specified by the user:

- `exposure_time`: The exposure time of the camera used for the measurements, in units of seconds.
- `calibration_path`: The path to the Picasso-generated .hdf5 file containing the picks selected for calibration, i.e. picked single binding sites.
- `data_path`: The path to the Picasso-generated .hdf5 file containing the picks selected for analysis, i.e.
- `dbclusters_path`: *Optional*, the path to the Picasso-generated .hdf5 file containing the convex_hull cluster areas.
- `ignore_dark`: The dark time, in units of frames, to ignore when linking localizations. This aids in separating photophysical blinking from DNA binding and unbinding kinetics.
- `prompt_cutoff`: True or False. When set to True, the user may specify a range of mean dark times to analyze.
- `title_0`: Overall title to be displayed over the plots.
- `title_1`: Title to be displayed over the calibration dataset plots.
- `title_2`: Title to be displayed over the counted data plots.
- `output_location`: Path to the folder where the plots and data are to be saved. Must end with a dash.
- `save_everything`: True or False. Set to true if all plots and data tables are to be saved.

After these parameters have been defined, the calibration may be performed. This is done by the function **Calibrate**, with the parameters:

- `screen_picks`: True or False. If set to True, a mean frame analysis is performed.
- `std_factor`: Factor by which the standard deviation of the mean frame is multiplied to select spots to omit.
- `std_cutoff`: Lower cut-off, in units of frames, for the standard deviation of individual spots.
- `show_screening`: True or False. If set to True, the mean frame analysis is displayed to the user.

Calibrate performs mean dark time fits on all spots in the `calibration_path` file (that pass the standard deviation filter if the filter is enabled). A plot showing the traces for each spot (in grey) and their respective best fits (in orange) is displayed to the user. The calculated mean dark times are represented with vertical purple lines. This gives the user visual confirmation that the fits have been performed successfully. If `prompt_cutoff` has been set to True, the user may then specify a range of dark times to continue analyzing, thus manually excluding poor fits. If `save_everything` is set to True, the calculated mean dark times for each spot are saved in a .csv file. The arithmetic mean of all calculated mean dark times is calculated and used for subsequent qPAINT analysis.

The function **Count** takes the same input parameters as Calibrate. It performs the (optional) standard deviation filter and calculates the mean dark times for all spots in the data_path file. It also displays the fit results for visual confirmation.

The function **Discuss**, which requires no additional parameters, displays a histogram of all calculated qPAINT cluster sizes (numbers of binding sites per cluster). If save_everything is set to True, all qPAINT results are saved in a .csv file. The group numbers (the numbers identifying each cluster in Picasso) are preserved.

DiscussWithSizes performs the same steps as Discuss, but additionally requires a dbclusters_path file. It outputs a .csv file containing each cluster, the qPAINT calculated number of binding sites as well as the convex_hull cluster area. This allows for consistency checks of the data.

Finally, **SpatialPlot** displays all identified clusters at their respective xy-positions. The height of the bars represents the qPAINT calculated cluster size. By setting cluster_size_cutoff, a .yaml file containing cluster locations for clusters larger than the specified size is exported. This file can be loaded into Picasso Render to manually inspect all large clusters, providing an additional verification method.

4. Function Overview

A. Linking localizations and calculating the mean dark times:

DetermineBursts

Input: All localizations from one group and the dark time to ignore.

Task: Identifies bright segments while taking ignore_dark into account.

Output: A list containing all bright segments ("bursts").

RemoveShortBrights (not applied in this publication)

Input: Bursts, ignore_bright.

Task: Removes bright segments shorter than the specified value.

Output: Bursts without short bright segments ("new_bursts").

CalculateBright (not applied in this publication)

Input: Bursts.

Task: Calculates the length of bright segments.

Output: Bright times.

CalculateDark

Input: Bursts.

Task: Calculates the length of the dark segments between the "bursts".

Output: List of all dark times for a given spot.

LinkLocalizations

Input: All localizations from one group, the dark time to ignore, the bright time to ignore.

Task: Sequentially performs DetermineBursts, RemoveShortBrights, CalculateBright and CalculateDark.

Output: Bright times, dark times and bursts for a given spot.

B. Filtering

MeanFrame

Input: Bursts

Task: Calculates the mean frame and standard deviation of the mean frame for a given spot.

Output: Mean frame and standard deviation thereof for a given spot.

ScreenLocs

Input: File name, a table containing all bursts for a given dataset (multiple spots), whether or not the screening should be displayed to the user, what multiple of the standard deviation to use for the mean frame filter, what number of frames to use as cutoff for the individual standard deviation filter.

Task: Performs a mean frame analysis and mean frame standard deviation analysis (as described in the publication and Wade *et al.* (2019)). If desired, the plots are displayed to the user.

Output: Indices for groups (picks) that pass the filters.

C. Fitting

expfunc

Input: Time t , mean dark time τ .

Task: Exponential function required to fit the cumulative distribution functions of the dark times for each spot.

Output: Function.

D. qPAINT Class

init

Input: As described in section 3 (instructions)

Task: Creates a class for the measurement to be analyzed.

CalcDark

Input: Path to file to be calculated, title to be displayed, whether or not to filter picks ("screen_picks"), the factor for the mean frame filter standard deviation, the standard deviation lower cutoff, whether or not to display the screening to the user, lower and upper dark time cutoff optionally specified by the user.

Task: Calculates the mean dark time for each group by conducting exponential fits for the CDF of the dark times of each group. First, it links localizations (see above). Optionally, the localizations are filtered. Dark times are calculated using expfunc. The fits are displayed to the user for visual verification.

Output: A pandas dataframe containing the spot numbers and respective mean dark times.

Calibrate

Input: Whether or not to filter picks ("screen_picks"), the factor for the mean frame filter standard deviation, the standard deviation lower cutoff, whether or not to display the screening to the user.

Task: Calls CalcDark for the calibration data set and displays the mean dark time histogram to the user. Also calculates the arithmetic mean of the mean dark times.

If prompt_cutoff has been set to true when creating the class, the user is additionally given the option to exclude dark times below and above given thresholds after having the preliminary results displayed to them. If they chose to, the calculations are repeated within the specified range.

Output: Arithmetic mean of mean dark times ("mean_dark").

Count

Input: Whether or not to filter picks ("screen_picks"), the factor for the mean frame filter standard deviation, the standard deviation lower cutoff, whether or not to display the screening to the user.

Task: Calls CalcDark for the data set to be counted. Calculates the number of binding sites based on the calibration mean of mean dark times.

If prompt_cutoff has been set to true when creating the class, the user is additionally given the option to exclude dark times below and above given thresholds after having the preliminary results displayed to them. If they chose to, the calculations are repeated within the specified range.

Output: Pandas table containing the qPAINT-calculated number of binding sites and xy-positions for each group.

Discuss

Task: Displays a histogram of qPAINT-calculated numbers of binding sites per group. If save_everything is set to True, the qPAINT results are saved to a .csv file.

DiscussWithSizes

Task: Displays a histogram of qPAINT-calculated numbers of binding sites per group.

Additionally reads the .hdf5 file specified in dbclusters_path to extract the Picasso-calculated cluster area ("convex_hull"). The optionally saved .csv file contains the qPAINT results and the convex_hull sizes of the respective groups (clusters).

SpatialPlot

Input: Whether or not to display the plot, lower limit of numbers of binding sites per cluster to be exported for further analysis ("cluster_size_cutoff").

Task: Displays a 2D bar plot where the x and y position of each bar represent the location of the cluster and the bar height (z) represents the qPAINT-calculated number of binding sites in that given cluster. If save_everything is set to True, a .yaml file containing the xy-position of all clusters larger than the number specified via cluster_size_cutoff is saved. This file can be loaded into Picasso render to highlight the clusters that require further inspection.