Documentation

Software Package

Unscrambling Fluorophore Blinking for Comprehensive Cluster Detection via Photoactivated Localization Microscopy

Rene Platzer[1], Benedikt K. Rossboth[2], Magdalena C. Schneider[2], Eva Sevcsik[2], Florian Baumgart[2], Hannes Stockinger[1], Gerhard J. Schütz[2], Johannes B. Huppa[1] and Mario Brameshuber[2]

[1] Institute for Hygiene and Applied Immunology, Center for Pathophysiology, Infectiology and Immunology, Medical University of Vienna, Vienna, Austria

[2] Institute of Applied Physics, TU Wien, Vienna, Austria

Version 1.0

Contents

All software was tested with Matlab 2017b (9.3.0.713579) and 2019b (9.7.0.1190202) running under Windows 10 (version 1809) and macOS Catalina (version 10.15.4).

1 Channel Registration

```
[tform] = main_channelRegistration( filepathCh1,filepathCh2 )
```

`main_channelRegistration` is the main function for calculating an affine transformation for the registration of two imaging channels based on localizations of fiducial markers detectable in both channels (e.g. TetraSpeck$^{\text{TM}}$ Microspheres, Thermo Fisher Scientific). Channel 2 may in the following be registered to channel 1 by applying the calculated transformation on localizations of channel 2. All values are assumed to be given in nm.

As a first step, corresponding localizations in the two channels are matched. The resulting localization pairs are then used to determine an affine transformation between the channels.

Input

The function allows for the following optional input arguments:

- `filepathCh1`: folder path to csv-files for channel 1, provided as string
- `filepathCh2`: folder path to csv-files for channel 2, provided as string

If the function is called without input arguments, the user is asked to select the file paths via a file selection dialog box.

The localization data must be provided as csv-files, with the first row being a header. The csv-files must contain columns for the x- and y-coordinates given in nm. Corresponding headers may be termed `x_nm`, `x [nm]` or `pos_x` for the x-coordinate, and `y_nm`, `y [nm]` or `pos_y` for the y-coordinate. In case fiducial markers are recorded over multiple frames, the frame number needs to be given in a separate column of the csv-file with the header `frame`. Corresponding localizations will be matched separately for each frame.
Multiple files may be present for each channel. All csv-files in the specified folder will be used for the analysis.

Parameters

For finding corresponding localizations in the two channels, the following parameters can be set in the subfunction `find_pairs`:

- `searchRadius`: Maximum expected distance between paired localizations due to chromatic aberration and localization error (line 27, $\in \mathbb{R}^+$, default: 120 nm).
- `maxShift`: Absolute value of the maximum expected shift between the channels (line 28, $\in \mathbb{R}^+$, default: 4000 nm).

Output

The output of the function is:

- `tform`: affine2d object storing the calculated affine transformation for channel registration

Localizations of channel 2 may be registered to channel 1 by applying the calculated transformation as follows:

```
[ ch2_x_corr,ch2_y_corr ] = transformPointsForward( tform,ch2_x,ch2_y )
```

2 Blinking Analysis

```
[blink_dist,blink_data]
 = main_analyseBlinkStat( pathBlinkData,pathPlatformData,tform )
```

`analyseBlinkStat` is the main function for the analysis of fluorophore blinking behavior based on input data from SMLM recordings. As precondition, the labeling concentration used in the experiment needs to be sufficiently low, so that individual fluorescent dyes can be spatially distinguished. The program allows to restrict blinking analysis to blinking data co-localized with control data from a platform. All values are assumed to be given in nm.

As a first step, the signals from both color channels, i.e. the platform data and the blinking data, are registered. A density filter is applied to the platform data and platform signals with nearest neighbors closer than `radiusDensityFilter` are discarded.
If `selectROI` is set to `true`, all further analysis can be restricted to a user-selected region of interest. Fluorescent signals from individual labels are grouped via hierarchical agglomerative clustering. The program allows to set the options for the distance metric, linkage criterion, and cutoff of the resulting dendrogram into individual clusters. Default options are the Euclidean distance metric, the unweighted average distance (UPGMA) linkage criterion and 200 nm cutoff, which allows to reliably cluster localization signals from well-separated single molecules.
If `performColoc` is set to `true`, colocalization analysis between the cluster centers and the remaining platform signals is performed. Blinking data signals are regarded as colocalized if a platform signal is located within a radius of `searchRadiusColoc` from a cluster center. Only colocalized localization clusters are kept for further analysis.
As a final step, the blinking behavior of the fluorescent labels is analyzed, including the total number of detections of the same label, the frame of first appearance, on-times, off-times, the number of bursts and the number of gaps in between. Labels detected more often than the chosen threshold value `maxBlinks` are discarded as outliers.

Input

The function allows for the following optional input arguments:

- `filesBlink`: folder path to csv-files for blinking data, provided as string. Blinking data is presumed to be recorded over multiple frames.
- `filesPlatform`: folder path to csv-files for platform data, provided as string. It is assumed that only one signal from each platform is given. If platform data was recorded over multiple frames, the signals need to be merged beforehand (merge radius r_{merge} should be selected dependent on the mean localization precision σ_{loc}, i.e. $r_{\text{merge}} \approx 2\,\sigma_{\text{loc}}$).

- tform: affine2d object storing the calculated affine transformation for channel registration (see Section 1).

If the function is called with input for filesBlink only, blinking analysis is performed without prior co-localization analysis with any platform data. If the function is called without any input arguments or missing tform, the user is asked to select the missing input via a file selection dialog box. Co-localization with platform data allows to reduce background signals. Analysis without platform data should only be performed if background signals are known to be negligible.

The blinking and platform data must be provided as csv-files, with the first row being a header. The csv-files must contain columns for the x- and y-coordinates given in nm. Corresponding headers may be termed x_nm, x [nm] or pos_x for the x-coordinate, and y_nm, y [nm] or pos_y for the y-coordinate.
The frame numbers for blinking data needs to be stored in a separate column in the csv-files with the header frame.
Multiple files may be present for both blinking and platform data. All csv-files in the specified corresponding folders will be used for the analysis.

Options for hierarchical clustering are set by user input via a dialog box. Default options are the Euclidean distance metric, the unweighted average distance (UPGMA) linkage criterion, and cutoff at the distance of 200 nm for determining individual clusters.

Parameters and Options

The following analysis parameters may be set in the main file:

- maxBlinks: threshold for maximum number of blinks for an individual label, labels detected more often than the set value are discarded as outliers (line 56, $\in \mathbb{N}$, default: 4000).
- radiusDensityFilter: threshold for density filtering of platform data, platform signals with nearest neighbors closer than the set value are discarded (line 59, $\in \mathbb{R}^+$, default: 500 nm).
- searchRadiusColoc: threshold for colocalization analysis, signals are regarded as colocalized if they are closer than the set threshold value (line 62, $\in \mathbb{R}^+$, default: 500 nm).
- analysis_startframe: only signals with frame numbers of detection higher or equal to the specified start frame are considered in the further analysis (line 65, $\in \mathbb{N}$, default: 1).

Furthermore, the following options are available:

Analysis options

- performColoc: if set to true the program performs colocalization analysis between blinking and platform data (line 71, true/false, default: true).

- `selectROI`: if set to `true` the analysis will be restricted to a certain region of interest (ROI) that can be selected by the user (line 74, `true/false`, default: `true`). The ROI shape may either be a rectangle or polygon (line 76):
  - `'rectangle'`: lets user select a rectangular ROI (default).
  - `'polygon'`: lets user select a polygonal ROI.

Plotting options

- `parPlot.plotFigures`: if set to `true` figures are plotted (line 83, `true/false`, default: `true`).

  The plotted figures are the following:
  - Localization maps of input platform data.
  - Density filtered platform data and blinking data.
  - Results of colocalization analysis.
  - Results of hierarchical clustering of localizations.
  - Histograms of determined blinking statistics (total number of detections, frame numbers of first appearances, on-times, off-times, number of bursts and gaps).
  - Time traces of detections of each label, ans box plot of the fraction of colocalized platform and blink signals.
- `parPlot.plotAll`: if set to `false` the plotting of figures is restricted to the first input file where applicable; otherwise figures are plotted for all input files (not recommended for a large number of input files) (line 84, `true/false`, default: `false`).

Saving options

- `parSave.save_blinkdist`: if set to `true` blinking data and resulting blinking statistics are saved, see also section Output (line 91, `true/false`, default: `true`).
- `parSave.save_figures`: if set to `true` figures are saved, see also section Output (line 94, `true/false`, default: `true`).

If any of the saving options is set to `true`, the user will be asked to select a path and name for saving the results and/or figures.

Output

The output arguments of the function are the following ones:

Variable	**Type**
`blink_dist`	struct
`blink_data`	struct array

`blink_dist` stores the blinking statistics in a structure with the following fields (m = total number of analyzed labels):

- `blink_dist.num`: total number of detections for each label, $m \times 1$ array
- `blink_dist.start`: frame of first appearance for each label, $m \times 1$ array
- `blink_dist.ton`: on-times, array of number of consecutive frames a label is detected during one burst
- `blink_dist.toff`: off-times, array of number of consecutive frames a label is in its dark state between two bursts
- `blink_dist.numBursts`: number of bursts for each label, $m \times 1$ array
- `blink_dist.numGaps`: number of gaps for each label, $m \times 1$ array

`blink_data` stores the localization data used to generate the blinking statistics. It is an array of structures storing the following fields for each input file of blinking data:

- `blink_data.filename`: filename of the input csv-file storing the blinking data
- `blink_data.blinks`: cell array of blinking data used for final analysis, containing the respective localizations and corresponding data for all analyzed labels of the respective input file.

Additionally, if the option to save results is selected (`parSave.save_blinkdist = true`), three files are generated:

- *`name`*`_blinkDist.mat` stores the resulting blinking statistics `blink_dist`.
- *`name`*`_blinkData.mat` stores the localizations data `blink_data.blinks` of all analyzed labels.
- *`name`*`_blinkData.csv` stores the localizations data of all analyzed labels in a csv-file.

If the option to save figures is selected (`parSave.save_figures = true`), the following figures are saved under the respective file name:

- *`name`*`_clustering_file1.png`: result of hierarchical clustering for first input file.
- *`name`*`_timetraces.png`: time traces of detections of each label.
- *`name`*`_histogramsBlinkDist.png`: blinking statistics histograms.
- *`name`*`_colocalization_file1.png`: result of colocalization analysis for first input file (available only if `performColoc = true`).
- *`name`*`_boxplotColocs.png`: box plots of colocalization analysis results
(available only if `performColoc = true`).

`name` can be specified as user input when running the main function.

3 Ripley’s K Analysis

```
[ rk_result_sample, rk_result_sim ]
 = main_ripley_w_blinks ( data, blink_statistics, density, runs )
```

main_ripley_w_blinks is the main function for performing Ripley’s K analysis on simulated and experimental data. Measured SMLM data may be compared with simulated SMLM data based on derived blinking statistics.

Input

The input parameters with their options are:

Variable	**Value**
data	[‘example’, ‘sample’, ‘none’]
blink_statistics	[‘example’, ‘sample’, ‘none’]
density	> 0
runs	≥ 5

data specifies the type of SMLM data to analyze:

- ‘example’: analyzes exemplary SMLM data (from a T cell expressing CD3z-PS-CFP2).
- ‘sample’: lets the user select SMLM data via a file selection dialog box.
- ‘none’: no SMLM data is analyzed, only the simulation is executed and resulting data analyzed.

The SMLM data must be provided as csv-file, with the first row being a header. The csv-file must contain columns for the x- and y-coordinates given in nm. Corresponding headers may be termed x_nm, x [nm] or pos_x for the x-coordinate, and y_nm, y [nm] or pos_y for the y-coordinate. The analysis can be restricted to a subset of the data in a certain region of interest (ROI) that can be selected by the user.

blink_statistics specifies the blinking behavior used for simulation:

- ‘example’: uses recorded blinking statistics of PS-CFP2.
- ‘sample’: lets the user select blinking data via a file selection dialog box.
- ‘none’: no blinking is simulated, only the molecular positions are simulated and analyzed.

The blinking statistic needs to be provided in a separate mat-file, which contains the structure blink_dist. The number of detections of each analyzed PA/PS-FP is stored as a column vector in the field blink_dist.num.

`density` specifies the molecular density used for the simulation. In case SMLM data is analyzed, the density is estimated from this data directly by dividing the amount of localizations by the average number of detections $\langle N \rangle$ per molecule. The value of $\langle N \rangle$ is derived from the selected `blink_statistics`.

`runs` specifies the number of simulation runs.

Parameters

The following analysis parameters may be set in the main file:

- `steps`: vector of sample distances r for which Ripley's K analysis is performed (line 9, $r \in \mathbb{R}^+$, default: $[1, 5, ..., 750]$). Number of steps may be reduced in order to decrease run time of analysis. Minus sampling is used as a method for edge correction in the calculation of Ripley's K statistics, leading to a reduction of analysed data points for high values of r.
- `roi`: side length of quadratic region of interest for simulation given in nm (line 10, $\in \mathbb{R}^+$, default: 5000).
- `pa`: localization precision, given as a structure containing the following fields:
  - `pa.mu`: mean localization precision in nm (line 13, $\in \mathbb{R}^+$, default: 30).
  - `pa.std`: standard deviation of localization precision in nm (line 14, $\in \mathbb{R}^+$, default: 5).
  - `pa.lo`: lower bound for localization precision in nm (line 15, $\in \mathbb{R}^+$, default: 10).
  - `pa.up`: upper bound for localization precision in nm (line 16, $\in \mathbb{R}^+$, default: 60).

Output

The output arguments of the function are the following ones:

Variable	**Type**
`rk_result_sample`	4 x steps matrix
`rk_result_sim`	runs x steps matrix

`rk_result_sample` stores the obtained Ripley's statistics for the SMLM sample. With the first row containing the analyzed distances r, the second row being the corresponding Ripley's K statistics, the third row Ripley's L statistics and the forth row $L(r) - r$.

`rk_result_sim` stores the values $L(r) - r$ of all simulation runs. Each row corresponds to one simulation.

Additionally, three files are generated:

- `*name*_loc_maps.mat` stores all the simulated localizations.
- `*name*_rk_results.mat` stores all the information relevant for Ripley's K analysis.
- `*name*_results.fig` stores the resultant figure. The figure consists of the following sub-panels:
  (A) ROI of experimental SMLM data (in case experimental input data is given).
  (B) Exemplary simulated data with ROI of same size as for experimental input data.
  (C) Results of Ripley's K analysis for experimental and simulation data.

name can be specified in line 20 in the main function.