sptPALM data analysis (Python & ImageJ)

Basic Markdown syntax: [link], Export to PDF

Workflow and code for single-particle, single-cell tracking analysis.

Content

1. Literature

1. "Evaluating single-particle tracking by photo-activation localization microscopy (sptPALM) in Lactococcus lactis." S.P.B. van Beljouw, S. van der Els, K.J.A. Martens, M. Kleerebezem, P.A. Bron, J. Hohlbein, Physical Biology, 16, 035001, 2019, [link]
2. "Visualisation of dCas9 target search in vivo using an open-microscopy framework." K.J.A. Martens, S. van Beljouw, S. van der Els, J.N.A. Vink, S. Baas, G.A. Vogelaar, S.J.J. Brouns, P. van Baarlen, M. Kleerebezem, J. Hohlbein, Nature Communications, 10, 3552, 2019, [link]
3. “Live-cell imaging reveals the trade-off between target search flexibility and efficiency for Cas9 and Cas12a”, L. Olivi, C. Bagchus, V. Pool, E. Bekkering, K. Speckner, H. Offerhaus, W.Y. Wu, M. Depken, K.J.A Martens, R. Staals*, J. Hohlbein*, Nucleic Acids Research, 52, 5241, 2024, [link]
4. "Direct visualization of native CRISPR target search in live bacteria reveals Cascade DNA surveillance mechanism", J.N.A. Vink, K.J.A. Martens, M. Vlot, R.E. McKenzie, C. Almendros, B. Estrada Bonilla, D.J.W. Brocken, J. Hohlbein, S.J.J. Brouns, Molecular Cell, 77, 39-50.e10, 2020, [link]
5. "Extracting transition rates in particle tracking using analytical diffusion distribution analysis (anaDDA)", J. Vink, S.J.J. Brouns, and J. Hohlbein, Biophysical Journal, 119, 1970–83, 2020, [link]

2. Known issues

1. --

3. Experimental and computational workflow

1. Run experiments on the **miCube** to obtain:
    1. single-molecule data, e.g., 'MyRawData.tif', recorded using single-particle-tracking photoactivaltion light microscopy (sptPALM).
    2. cell outlines, e.g., 'MyBrightfield.tif', recorded using bright field microscopy.

2. run script called *Macro_thunderSTORM.ijm* in **ImageJ/Fiji** on 'MyRawData.tif' to obtain 'MyRawData_thunder.csv' which contains the localisations of the fluorophores. Details: ImageJ / Fiji: segmentation and localisation.
3. run script called *Macro_CellSegmentation.ijm* in **ImageJ/Fiji** on 'MyBrightfield.tif' to obtain 'MyBrightfield_procBrightfield.tif', 'MyBrightfield_procBrightfield_segm.tif', and 'MyBrightfield_procBrightfield_segm_Table.csv' containing segmented cells and associated data. Details: ImageJ / Fiji: segmentation and localisation.

4. ImageJ/Fiji

We run two macros *Macro_thunderSTORM.ijm* and *Macro_CellSegmentation.ijm* to prepare the raw data for subsequent analysis. There are a number of parameters set in the functions that might have to be adjusted depending on the specific circumstances of your measurements. When in doubt, ask! To open and later run the macro, drag & drop each macro into **ImageJ/Fiji**. To have all required (sub-)functions available in **ImageJ/Fiji**, check the following

1. -> ImageJ/Fiji -> Help -> Update... -> Manage Update sites -> tick the box for 'HohlbeinLab' and 'SCF MPI CBG'
2. restart ImageJ/Fiji

4.1 *Macro_CellSegmentation.ijm*

Macro_CellSegmentation.ijm is used to segment cells on 'MyBrightfield.tif'. Currrently, the macro runs trough the following steps and allows selecting more than one file.

1. Load raw data 'MyBrightfield.tif' using a GUI or providing a filename in the script.
2. Run function for z-projection, averaging over all frames to reduce noise
3. Further data post-processing for stability of cell segmentation
4. Save processed brightield image 'MyBrightfield_procBrightfield.tif' in current folder
5. Run segmentation and save segmented image 'MyBrightfield_procBrightfield.tif' and labels 'MyBrightfield_procBrightfield_segm_Table.csv' in the current folder

4.2 *Macro_thunderSTORM.ijm*

Macro_thunderSTORM.ijm is used to perform the subpixel localisation of single-molecule data 'MyRawData.tif'. Currrently, the macro runs trough the following steps

1. Load raw data 'MyRawData.tif' using a GUI or providing a filename in the script
2. Remove first 500 frames to prevent attempted localisation of overlapping fluorophores
3. Run FTM2 (fast temporal medium filter 2). More information here [Github FTM2].
4. Run thunderSTORM-phasor to localise all fluorophores and save data in the form of a CSV table named 'MyRawData_thunder.csv'. ThunderSTORM can be run either with 'phasor' (faster) or 'MLE' (slower but allows filterring for PSF widths')

5 Python

Python code to analyse experimental data and simulate single-particle tracking in bacteria. Make sure that you have a

1. Python installation (e.g., Anaconda)

2. Python development environment (e.g., Spyder)
3. trackpy installed (to install: e.g., 'pip install trackpy')

5.1 sptPALM_main.py

Main function to analyse experimental data. We require:

1. *.csv file(s) containing the x,y and, optionally, z positions of individual emitters as obtainbed via, for example, ThunderSTORM or any other SMLM data suite.
2. (optional) brightfield images of cells, their segmentation, and a *.csv table containing relevant information on the segmentation

Run the function in the command line of your Python development environment by typing: *runfile('/...your folder.../GitHub/sptPALM-Python/sptPALM_main.py', wdir='/...your folder.../GitHub/sptPALM-Python')* and pressing Enter.

The following prompt will appear:

```
0: Exit
1: Set parameters GUI
2: Analyse individual movies
3: Combine individually analysed movies
4: Plot combined data
5: Monte-Carlo DDA
6: Auxillary functions
```

Option	Description
0: Exit	Closes the prompt and returns to the command line.
1: Set parameters GUI	Runs *set_parameters_sptPALM.py* loading default settings for the data analysis followed by *set_parameters_sptPALM_GUI.py* that allows changing, loading, and saving specific sets of parameters to analyse the experimental data
2: Analyse individual movies	Runs *analyse_movies_sptPALM.py* to analyse individual movies as earlier defined option 1. If no output from 1 is in memory, the option 1 is run again. Returns a DataFrame called 'data' which contains all localisations and further information. Results are saved into '**sptData_movies.pkl**' or similar
3: Combine individually analysed movies	Runs *Combine_individually_analysed_movies.py* to group the data of individually analysed files available in the DataFrame 'data' based on conditions defined in *set_parameters_sptPALM.py*. If no data is in memory, a GUI will open to load '**sptData_movies.pkl**'' or similar. Function returns a DataFrame called 'comb_data'. Results are are saved into '**sptData_combined_movies.pkl**' or similar

Option	Description
4: Plot combined data	Graphical output of the data combined in Option 3. Name of function: plot_combined_data_sptPALM.py
5: Monte-Carlo DDA	Runs *MC_diffusion_distribution_analysis_sptPALM.py* to perform fitting of the experimental data based on parameters defined in *set_parameters_simulation.py*
6: Auxillary functions	Provides option to combine *.csv files to enable particle counting per cell over many movies

5.2 Subfunctions called in *analyse_movies_sptPALM.py*

Name	Description
load_localisations_from_csv.py	Imports *.csv data (e.g., from running ThunderSTORM) and extends the data frame with additional columns before saving it on the disc. Note that we change from unit [nm] to [µm]
apply_cell_segmentation_sptPALM.py	Using provided segmentation data to filter and refine localisations
tracking_sptPALM.py	Links individual localistions to tracks. We use trackPy based on [link] written by Crocker and Grier. The output of track.m has the format: 'x [um]', 'y [um], 'frame', 'particle', the latter of which we rename to 'track_id'. If segmentation was used, only localisations within individual cells are considered to form a track
analyse_diffusion_sptPALM.py	Calculates the apparent diffusion coefficient for all tracks provided by 'tracking_analysis.py' that have a certain minium and maximum number of localisations per track. Careful, diffusion coefficient is calculated using single steps and not classical mean square displacement over different step sizes!
plot_diffusion_tracklengths_sptPALM.py	Plots histogram of diffusion coefficients and histogram of track lengths
single_cell_analysis_sptPALM.py (optional)	Function that links tracks and diffusion coeficients to individual cells. Also later abbriviated as SCTA
plot_single_cell_analysis_sptPALM.py (optional)	Function plotting several things (bee swarm plots) based on cell by cell analysis.

5.1 simulation_main.py

To simulate distributions of diffusion coefficients, the following functions are run after each other:

Name	Description

Name	Description
set_parameters_simulation.py	Function for setting all parameters
initiate_simulation.py	Function for setting all starting positions, starting states etc
diffusion_simulation.py	Function for moving particles and checking for state changes
diff_coeffs_from_tracks_fast.py	Function to calculate diffusion coefficients for different track lengths
plot_diff_histograms_tracklength_resolved.py	Function for plotting the data

5.3. Settings defined in '*set_parameters_sptPALM.py*'

SCTA: Single-cell tracking analysis. Parameters in order of appearance.

name	Default	Description
data_dir	-	Directory where the data is found (selection via GUI or pre-defined, see below)
default_output_dir	output_python/	Initialise new directory to which analysed data is saved
fn_locs	-	Initialise filename to the localisation data
fn_proc_brightfield	-	Initialise filename to the brighfield data
fn_csv_handle	_py_out.csv	Will be used to name the csv file of the analysed data
fn_dict_handle	_py_out.pkl	Will be used to name the pickle file of the analysed data
fn_diffs_handle	_diff_coeffs.csv	Will be used to name the file of diffusion coefficients
fn_movies	sptData_movies.pkl	Will be used to name the pickle file of the analysed data
fn_combined_movies	sptData_combined_movies.pkl	filename of combined conditions output
condition_names	[]	Initialise, further defined below
condition_files	[]	Will be used to name the pickle file of the analysed data

name	Default	Description
copynumber_intervals	[]	ilename of combined conditions output
pixelsize (µm)	0.119	Effective pixel size of the camera (also set in *Macro_thunderSTORM.ijm*). The number is close to the actual pixel size of the camera (Photometrics 95B: 15 µm, Andor Zyla 4+: 13 µm after 2x2 binning) divided by the magnification of the objective 100x (Nikon, N.A. = 1.49)
cellarea_pixels_min	50	Filter cells for minum area (area is given in number of pixels), default: 50
cellarea_pixels_max	300	Filter cells for area (area is given in number of pixels), default: 500
use_segmentations	True	Segmentation of cells allows linking localisations to individual cells
track_steplength_max (µm)	0.5	Used for tracking. Maximum distance that two localisations in consecutive farmes can be apart to still be considered belonging to a single track. Avoids linking of non-related localisations. Note that that this thresholds can effect the histogram of diffusion coefficients
trackMemory	0	Set tracking memory in frames. Allows tracks to have missing localisations. default 1
frametime	0.01	Length of a single movie frame in seconds
loc_error (µm)	0.035	Assumed localisation error (µm). Could be calculated based on the localisation precisionin ThunderSTORM, but not possible with phasor-based localisation.
diff_hist_steps_min	3	Minimum number of steps for a track to be analyzed. Note that the actual number of localisations is one higher

name	Default	Description
diff_hist_steps_max	100	Minimum number of steps for a track to be analyzed.
track_lengths	1,2,3,4,5,6,7,8	Minimum number of steps for a track to be analyzed.
number_tracks_per_cell_min	1	Minimum number of tracks that each cell must contain
number_tracks_per_cell_max	10000	Maximum number of tracks that each cell must contain
scta_vis_cells	False	Visualize individual cells true/false
scta_plot_cell_window	15	Radius in pixels for plotting individual cells and their tracks
scta_vis_interactive	False	Interactively cycle through indiviudally plotted cells true/false
scta_vis_rangemax	0.3	Color-coding in the range of 0 to (SCTA_vis_rangemax * plotDiffHist_max)
plot_diff_hist_min	4E-3	plot and histogram from min $\mu m^2/s$ to max $\mu m^2/s$
plot_diff_hist_max	10	plot and histogram from $\mu m^2/s$ to $\mu m^2/s$
binwith	0.1	width of bins, default: 0.1
fontsize	10	Fontsize used to plot figures
linewidth	1	linewidth used to plot figures
plot_norm_histograms	'probability'	Choose style of plotting some histograms. Optiomns: 'count', 'probability', 'countdensity', 'pdf', cumcount', 'cdf'
plot_frame_number	True	whether to plot the frame numbers next to the tracks in 'Plot_SingleCellTrackingAnalysis.m'
dpi	150	DPI setting for plotting figures, default: 300
cmap_applied	gist_ncar	color map for segmented cells was: 'nipy_spectral', tab20c,

5.4. Variables internally used in the Python analysis

Name	Default	Description
data_dir	**User defined**	Directory containing your data
default_output_dir	**User defined**	Directory to which new data is saved. Default: 'output_python/'
fn_locs	**User defined**	List of one or more '_thunder.csv' files to be analysed
fn_proc_brightfield	**User defined**	List of one or more processed brightield images for cell segmentation '_procBrightfield.tif'. Filename is also used to locate the segmented image and corresponding *.csv table!
csv data saved as '_py_out.csv'	initialised in *load_csv.py*	Columns: loc_id, movie_id, frame_id, track_id, cell_id, x [µm], y [µm], z [µm], brightness, bg, i0, sx, sy, empty (NaN). CSV table based on the structure of the CSV output provided by ThunderSTORM (software used for sub-pixel localisation)
tracks	--	Some 'tracks' have a length of one thereby representing single localisations! Columns: x(µm), y(µm), time(frame), track_id. If 'useSegmentations = true', 'TrackingAnalysis.m' runs for each valid cell (cellarea_pixels min/max thresholds active) using all localisations found therein. **Not filtered for 'diff_hist_steps_min(max)' and 'number_tracks_per_cell_min(max)'!**
DiffsCoeffsList	--	List of diffusion coefficients (unit: µm^2/s!) and more. Columns: DiffCoeffsFiltered, #localisations, track_id, cell_id and copynumber. **Filtered for 'diff_hist_steps_min(max)'**
scta_table	--	Overview table after filtering for 'DiffHistSteps', 'numberTracksPerCell' in valid cells (CellAreaPix_min/max thresholds active). Columns: cell_id, #locs per cell, cumulative #locs, #tracks (filtered for #tracks per cell), cum. #tracks (filtered for #tracks per cell), cum. #tracks (unfiltered for #tracks per cell), keep_cells, averageDiffCoeffperCell
scta_tracks	--	Table containg for each valid cell, the full information for each localisation of each valid track

Name	Default	Description
tracks_filtered	--	Columns: x(µm), y(µm), time(frame), track_id. **Filtered for 'diff_hist_steps_min(max)s' and 'number_tracks_per_cell_min(max)'! using valid cells**