OneFlowTraX user guide

Sven zur Oven-Krockhaus, Center for Plant Molecular Biology (ZMBP) / Institute for Physical and Theoretical Chemistry, University of Tübingen (sven.zur-oven-krockhaus@uni-tuebingen.de)

Table of Contents

1 Purpose and short description

OneFlowTraX is a software suite for analyzing single molecule localization microscopy (SMLM) data. It is particularly targeted for its use in the analysis of dynamics and cluster formation in the plasma membrane of plants and was developed to provide a user-friendly graphical user interface (GUI) to perform typical steps of SMLM analysis: (i) localization, (ii) tracking, (iii) mobility analysis, and (iv) cluster analysis. The parameters and visualized results of each step are handled in self-contained tabs, allowing the user to check the data quality for each step and adjust the parameters before sending the data to the next step in a linear manner. After verifying the parameters for the relevant steps in the analysis, e.g., for a representative subset of the data, a batch processing utility can be used to read in lists of input files that will be analyzed automatically using the chosen parameters. As data integrity is often a major concern with advanced data analysis using a multitude of parameters, this makes sure that all relevant data (the chosen settings and the calculated results) are consolidated in one results document that can be stored together with the SMLM raw data.

2 Installation

2.1 Installation of the software

Please download the latest version of this software (including this manual) from GitHub (https://github.com/svenzok/OneFlowTraX) by clicking "<> Code" and "Download ZIP". Extract all files to a folder of your choice.

2.1.1 Installation as a stand-alone software

Double-click the OneFlowTraX_Installer.exe file and follow the instructions of the installer. You will also be instructed to download and install the MATLAB Runtime (around 8 GB) that is necessary for the execution of the app, so make sure that you are connected to the internet. Alternatively, see https://www.mathworks.com/products/compiler/matlab-runtime.html to download and install the Runtime independently. When the OneFlowTraX installation has finished, you can start the program by navigating to the install folder and double-click the OneFlowTraX.exe file in the "application" sub-folder. On Windows platforms, the program will also appear in the Start Menu and can be started from there.

2.1.2 Installation as an executable app in MATLAB

To install OneFlowTraX as a MATLAB app, double-click the OneFlowTraX.mlappinstall file. It will then appear as an app in the Apps ribbon (under MY APPS) and can be started by double-clicking its icon.

To run inside MATLAB, OneFlowTraX requires additional MATLAB products:

- Signal Processing Toolbox 9.2
- Image Processing Toolbox 11.7
- Statistics and Machine Learning Toolbox 12.5
- Parallel Computing Toolbox 7.8 (for GPU usage)

Alternatively, you can navigate to the respective main program folder in the MATLAB browser (right-click "Add to Path" > "Selected Folders and Subfolders") and start the app by double-clicking the OneFlowTraX.mlapp file or by typing `OneFlowTraX` in the MATLAB command window, followed by the ENTER key.

3 Requirements and Constraints

This software was developed with MATLAB Version 9.14.0.2206163 (R2023a) on a Microsoft Windows platform (Windows 10 Enterprise Version 10.0 (Build 19045), 64-bit), as such, it was only tested on computers with that architecture. As the single molecule localization procedure is the speed bottleneck of the analysis, a CUDA-compatible graphics card is highly recommended for a boost in processing speed (in our case: GeForce GTX 2070). You can check which NVIDIA® GPU architectures are currently supported by MATLAB under https://www.mathworks.com/help/parallel-computing/gpu-computing-requirements.html and check suitable graphics cards under https://developer.nvidia.com/cuda-gpus. The program will still work without such a graphics card, using instead the computer's main processing unit, but at a great loss of speed for the localization step (by a factor of 20 to 30). For the other analysis steps, an average CPU (in our case: AMD Ryzen 5 4500) is sufficient.

This program was designed as an all-in-one software pipeline, focusing on the most common SMLM analyses that are featured in current plant membrane protein dynamics literature. It aims to offer beginners in the field a more straightforward entry point into advanced data analysis techniques, with a focused approach on the most crucial parameters. Consequently, we chose to omit modifiable plug-ins for further or alternative analysis, but we strive to constantly develop the program and add new options to the core analysis steps according to the needs of the research community. Advanced users can, however, change the freely available code to their needs. Moreover, the batch analysis enables the output of intermediate results (localization lists, track lists), which can be used for calculations that are beyond the scope of this software.

4 Overview

You can start OneFlowTraX either in the MATLAB environment or by starting the compiled program (see section 2.1). After a few seconds of initialization, the main window will present as in Figure 1. Here you can access the main analysis steps with the tab bar (1), including Localization (the default starting tab), Tracking (which also includes the mobility analysis), Cluster Analysis and Batch Analysis. The following chapters are intended to make you more familiar with the individual tabs, analysis steps and visualizations. Note that only images can be saved in the analysis tabs, but not any data. The processing of a set of input files and the storage of all relevant analysis results is done entirely during batch analysis (section 4.4).

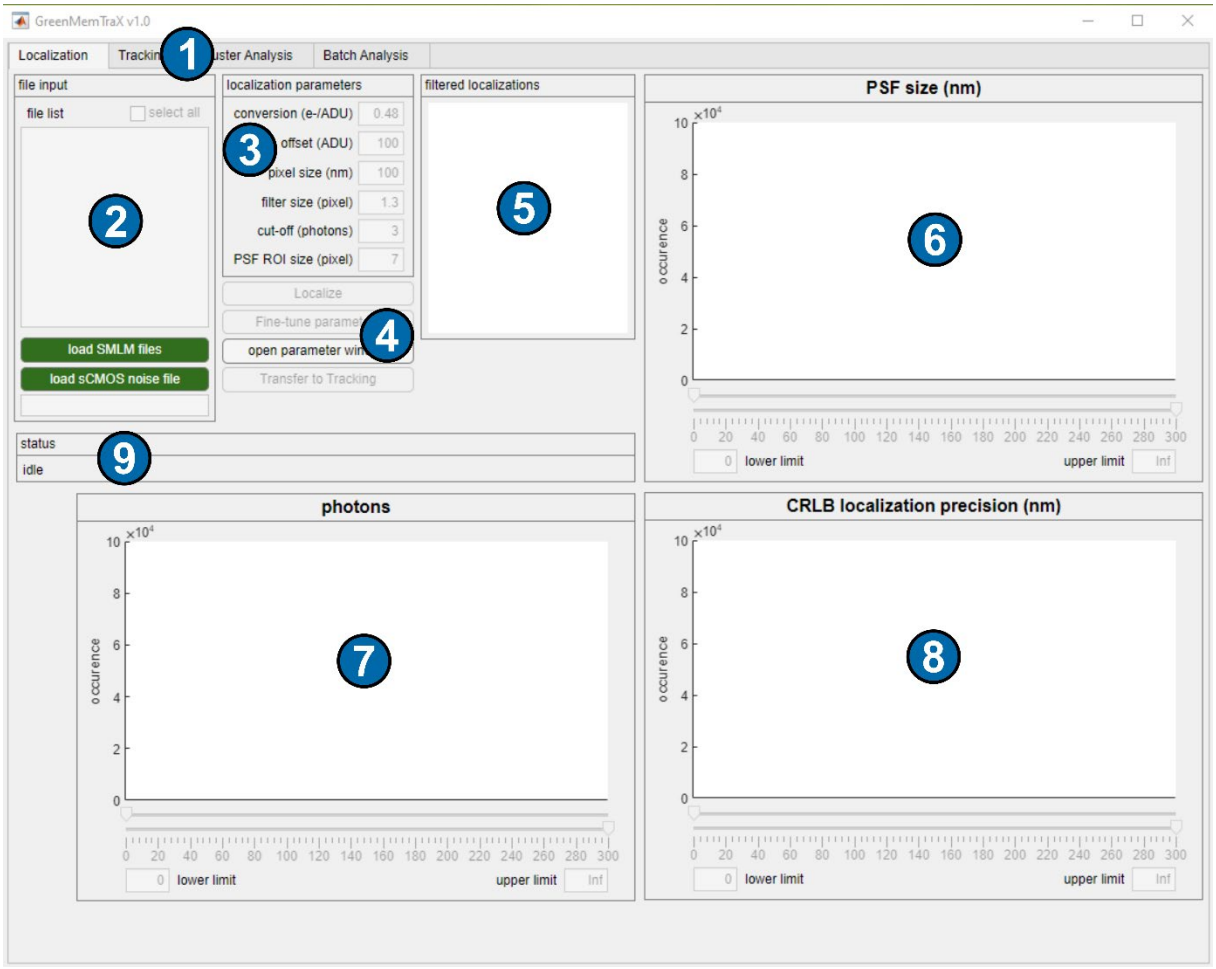

Figure 1. Localization tab of OneFlowTraX.

(1) tab bar; (2) file input panel; (3) localization parameters; (4) buttons for starting the localization, fine-tuning the localization parameters, opening the parameter window and transferring the data to the Tracking tab; (5) text window showing the number of results that were filtered out after the localization procedure; (6)-(8) histograms visualizing the results of the localization showing the PSF sizes, the number of photons and CRLB localization precision, (9) status panel, informing the user which file is currently being processed.

4.1 Localization tab

4.1.1 Loading SMLM raw data files

The addition of new files is handled in the file input panel (2). Pushing the button 'load SMLM files' will open a dialog window to choose one or more .TIF files. The new files will appear in the file list, while their full folder name can be seen as a tooltip by mousing over them. Files can only be loaded from one folder and adding new files by clicking 'load SMLM files' again will replace the old files in the list.

4.1.2 Loading a noise file

Most likely, the sptPALM data was acquired using a (scientific) complementary metal-oxide-semiconductor (sCMOS) camera, therefore you need to provide a noise file to correctly fit the data with an sCMOS noise model (Huang et al., 2013). By clicking 'load sCMOS noise file', a dialog window opens that lets you choose a corresponding .TIF image stack. This is easily acquired by collecting a series of 2 000-10 000 dark images (e.g., by screwing on the dust cap of the camera) with the same settings (frame rate, pixel dimensions and position on the camera chip) as the corresponding SMLM files. Without changing the acquisition parameters, they are usually valid for months, but we

recommend acquiring one noise file per experiment day (which also makes it easier to find the correct noise file for the data). After loading in the noise file, a variance map is calculated to determine the pixel-wise intensity variance of the selected camera chip region. Alternatively, a pre-calculated variance map can be provided (a .TIF file with only one frame) to circumvent this calculation. The text field below the button contains the name of the noise file, a tooltip also shows its folder.

For EMCCD cameras, the above corrections are not necessary for the analysis steps in this software. However, as OneFlowTraX expects a noise file, please use the provided 'EMCCDnoise.TIF' file (provided with this software), which will then be ignored during noise correction. This feature might change in a future release. While default localization parameters (compare section 4.1.3) are mainly based on sCMOS cameras, appropriate settings for EMCCD cameras can be quickly found using the fine-tuning window (see section 4.1.6).

4.1.3 Setting the localization parameters

The localization of fluorophore positions is highly dependent on the choice of the localization parameters in the localization parameters panel (3).

- offset (ADU): offset of the camera (without any light) in analog to digital units (ADU) that prevents negative values due to natural read noise fluctuations. This value can usually be found in the data sheet of the camera and is typically around 100. If you loaded in an image stack as a noise file, this value will be automatically calculated.
- conversion (e-/ADU): factor to convert ADU into photoelectrons (reciprocal of the average pixel gain). This value is usually given in the data sheet of the camera. The default value (based on a Hamamatsu Orca4.0 Flash v2 sCMOS camera) is 0.48.
- pixel size (nm): length in the sample plane that is projected onto one camera pixel in the detection plane. This value mainly depends on the overall magnification of the microscope and can be checked with appropriate calibration slides, e.g., fluorescent microspheres with known diameters. Ideally, the fluorescence spot of a single fluorophore should cover 2-3 pixels in the x- and y- dimensions to prevent under- or oversampling, resulting in a pixel size of ca. 100 nm.
- filter size (pixel): sigma of the first Gaussian in the difference of Gaussians (DoG) filter to enhance the contrast for the basic detection of bright spots in each image. A value between 1 and 1.5 usually delivers good results.
- cut-off (photons): minimal value for the brightest pixel in a spot to be included in the localization procedure, to prevent the unnecessary processing of background pixels and artifacts. The default value of 3 must be adjusted when either too many or too few fluorophores are found (see section 4.1.5), especially when deviating from the default value for the conversion factor (see above).
- PSF ROI size (pixel): width of the range of interest (ROI) square with which each eligible spot is cut out and processed further. Too small values may not capture the entire point spread function (PSF), too large values may capture more than one. This must be an odd number and 7-11 pixels are typically chosen for pixel sizes (see above) around 100 nm.

4.1.4 Starting the localization procedure

When one or more SMLM files and a noise file are loaded, the Localization button (4) turns green and can be pushed to start the procedure. Here, we adapted the core of the SMAP software (Ries, 2020) to perform the 2D localization. Its algorithm for determining single molecule positions uses a robust maximum likelihood estimation (MLE) of Gaussian PSF models, which was shown to be very

accurate, achieving the theoretical maximum given by the Cramer-Rao Lower Bound (CRLB). It also uses the graphics processing unit (GPU) of an installed compatible graphics card for outstanding processing speed. When no or no compatible GPU is found, the calculations will be performed by the central processor of the computer, albeit much slower. The status panel (9) informs about the current progress of the localization. After all files were processed, the results are presented in the filtered localization panel (5) and the results histograms (6)-(8), see below.

4.1.5 Interpreting the localization results

The filtered localization panel (5) gives a first impression on the success of the localization procedure.

- Original number of localizations: this value shows how many (putative) fluorophore spots were extracted from the selected sptPALM file(s) and forwarded to the fitting algorithm.
- including results with: this list counts the fitting errors. A localization is considered faulty, when any of its values is
    - nonsensical, like NaN ('not a number'), zero or an infinite value
    - an obvious outlier (>10x the data median)
    - stuck at a boundary condition and could not be further optimized by the fitter.
- Remaining (after pre-filtering): this is the absolute number of remaining localizations and their percentage based on the original number of localizations after the faulty localizations (see above) were removed. The absolute numbers usually do not sum up, as some localizations show multiple fitting errors.
- Remaining (after histogram filtering): this gives the absolute number of remaining localizations after pre-filtering (see above) and after applying any user-specified limits (see section 4.1.7). The percentages are based on either the original number of localizations or the pre-filtered ones.

The remaining localizations after pre-filtering should be above 90%, otherwise the quality of the raw data must be checked, or the localization parameters must be adjusted (see section 4.1.3). The above values are calculated for the currently selected files in the list. Individual files can be selected with the left mouse button (LMB), multiple ones with Ctrl+LMB, Shift+LMB, or all of them by checking the 'select all' box. This is also true for the histograms (6)-(8) that show the distribution of the values for the PSF size (6), number of photons (7) and CRLB localization precision (8) for the currently selected files. Ideally, these histograms should not deviate too much from normal (PSF size) or lognormal (photons) distributions, while the peak of the localization precision is below 30 nm (compare Figure 2).

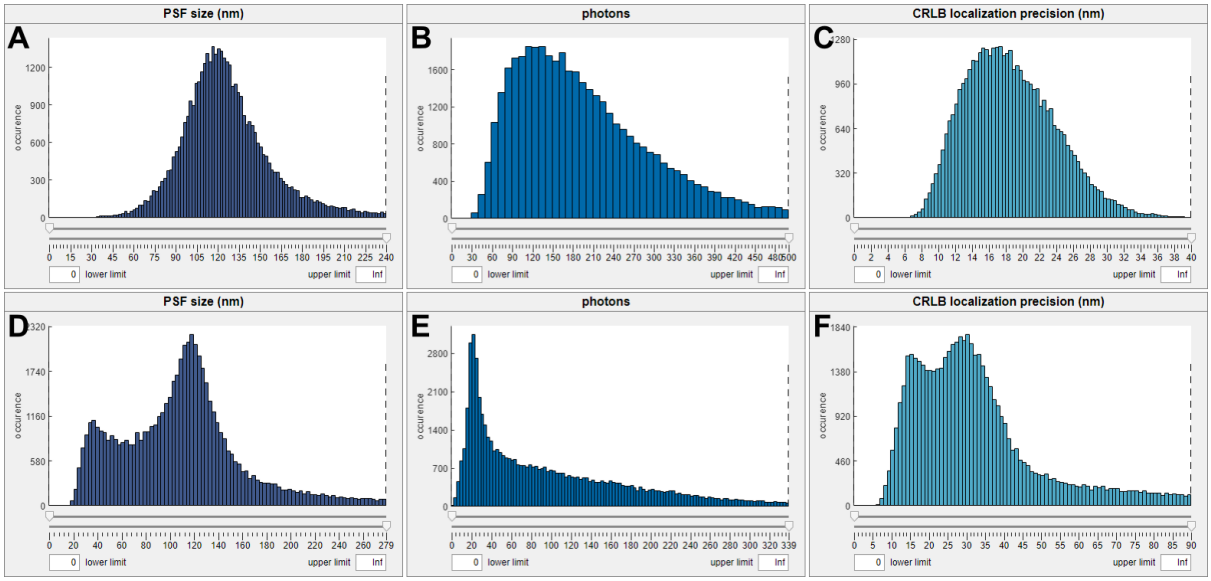

Figure 2. Examples of localization results.

A)-C) Typical histograms that result from well-localized molecule positions. D)-E) Suboptimal results show up in non-uniform distributions and other anomalies, e.g., due to unfavorable choice of localization parameters or poor quality of the raw data.

4.1.6 Fine-tuning the localization parameters

In case the localization procedure delivers unsatisfactory results (as seen in the filtered localization panel (5) or the histograms (6)-(8), also compare Figure 2), you can select a single file and then push the 'Fine-tune parameters' button (4). A helper window will open, where you can try out different parameter combinations for a chosen frame to improve the number of localized positions (or to suppress the number of background localizations). You can either close the window without changing anything or transfer the chosen parameters to the main program. Afterwards, you need to click the 'Localize' button again to repeat the localization procedure for all files with the new parameters.

4.1.7 User-defined filtering of the results

With super-resolution techniques like photo-activated localization microscopy (PALM) and stochastic optical reconstruction microscopy (STORM), localized emitter positions that persist over several frames need to be merged. Obviously, this is not done for sptPALM, since here individual emitters are to be tracked over several frames. However, it is common practice to filter SMLM data to improve data quality, e.g., to exclude defocused fluorophores or those whose positions could not be determined accurately. This can also be done here with the range sliders below the histograms (6)-(8), shown for the PSF size and photon numbers in Figure 3.

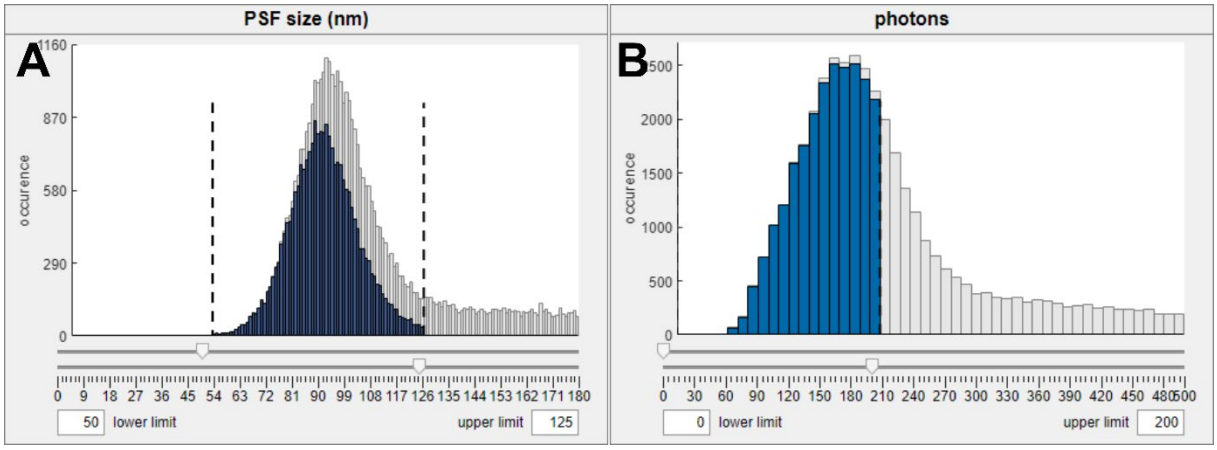

Figure 3. Histograms with user-defined limits in the Localization tab.

A) PSF size histogram, the user-defined limits (50-125 nm) are set via the edit fields or the slider knobs and are visualized by vertical dashed lines. B) Photon numbers histogram with the upper limit set to 200. All disregarded localizations are grayed out, which also has an effect on all other histogram (note that here, disregarding high photon numbers mainly affected the right shoulder of the main peak in A).

For each histogram, you have the option to define a range either by using the slider knobs or by manually entering values into the edit fields. The dashed lines visualize the limits in the graph and discarded data is shown as gray background. Note that the limits set for different histograms are interdependent. Figure 3 shows an exaggerated case, where higher photon numbers were discarded (Figure 3B), resulting in the removal of predominantly larger PSF sizes (Figure 3A), visible as the gray area on the right flank of the peak. One reason for this positive correlation of photon numbers with PSF sizes is that multiple overlapping fluorophores are detected as one. We suggest using this adjustment option carefully and not discarding too many fluorophore positions. The effect of the user-defined filtering can be seen in the filtered localization panel (5), showing the remaining localizations. If the recorded area contains fluorescence spots with a good signal-to-noise ratio, it is usually not necessary to discard more than an additional 5% of the data. For the subsequent tracking procedure, it is better to have some less accurately determined positions in a track than no track at all. Moreover, a region of interest for the tracks can be defined later (see section 4.2.6) to exclude localizations, e.g., from outside the cell membrane or defocused areas. To remove a limit, you can either set the lower limit to 0, the upper limit to Inf (write these values in the corresponding edit fields) or push the range slider knobs to their minimum/maximum values, which will update the edit field values as given above.

4.1.8 Setting parameters in the parameter window

One of the main features of OneFlowTraX is to provide a comprehensive overview of the currently set parameters for all analysis steps. This overview can be accessed by pushing the ‘open parameter window’ button (4) that is also present in all other tabs. This overview (Figure 4) can be closed and reopened at any time and lists all parameters that are currently set in the GUI. Clicking on any tree node (10), the corresponding parameters will be shown in the right panel (11) – shown here for the localization parameters that are also found in the localization tab under (3). Changing any parameter in the main program will also update this value in the parameter window. Conversely, you can also change a parameter in the parameter window (11), which will update the value in the corresponding tab of the main program. Note that this might reset the data analysis in OneFlowTraX to a point that was not dependent on that changed parameter, necessitating some recalculation. You can also load and save (12) entire parameter sets. In practice, you should load such a set at the beginning of the analysis.

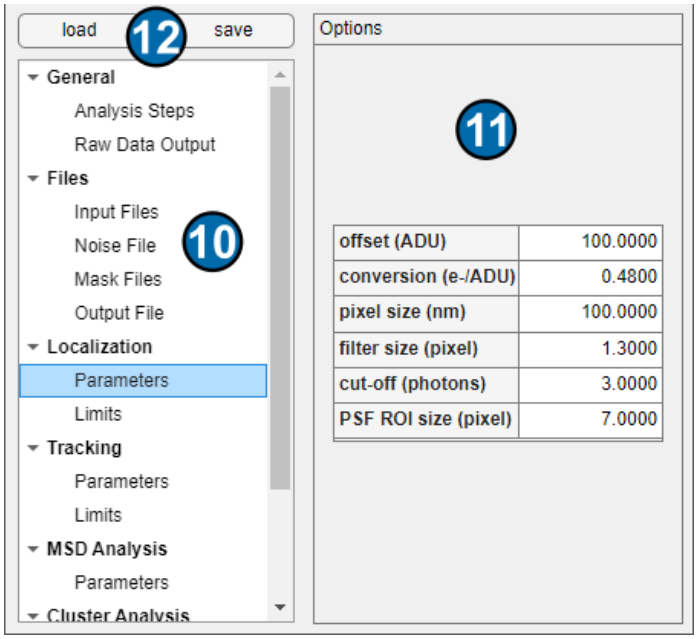

Figure 4. Parameter window to view, set, load and save all relevant parameters for the SMLM analysis steps in OneFlowTraX.

(10) Tree node selection area; (11) editable parameters for the chosen node; (12) buttons to load and save parameter sets.

4.1.9 Transferring the localization data to the Tracking tab

With localization data present, the 'Transfer to Tracking' button (4) turns green and allows to transfer the data of the currently selected files to the Tracking tab. Note that this will only forward those localizations per file that remain after filtering (compare the last line in the filtered localizations panel (5)). Pushing the button will automatically switch to the Tracking tab, but you can return to any tab at any time. Changing a parameter in a previous tab will, however, reset the analysis to that tab until the necessary calculations are repeated with that new parameter.

4.2 Tracking tab

This tab combines the track building step and mobility analysis as shown in Figure 5. When it is opened first (e.g., after transferring the localization data), only the 'Build tracks' and 'open parameter window' buttons will be enabled.

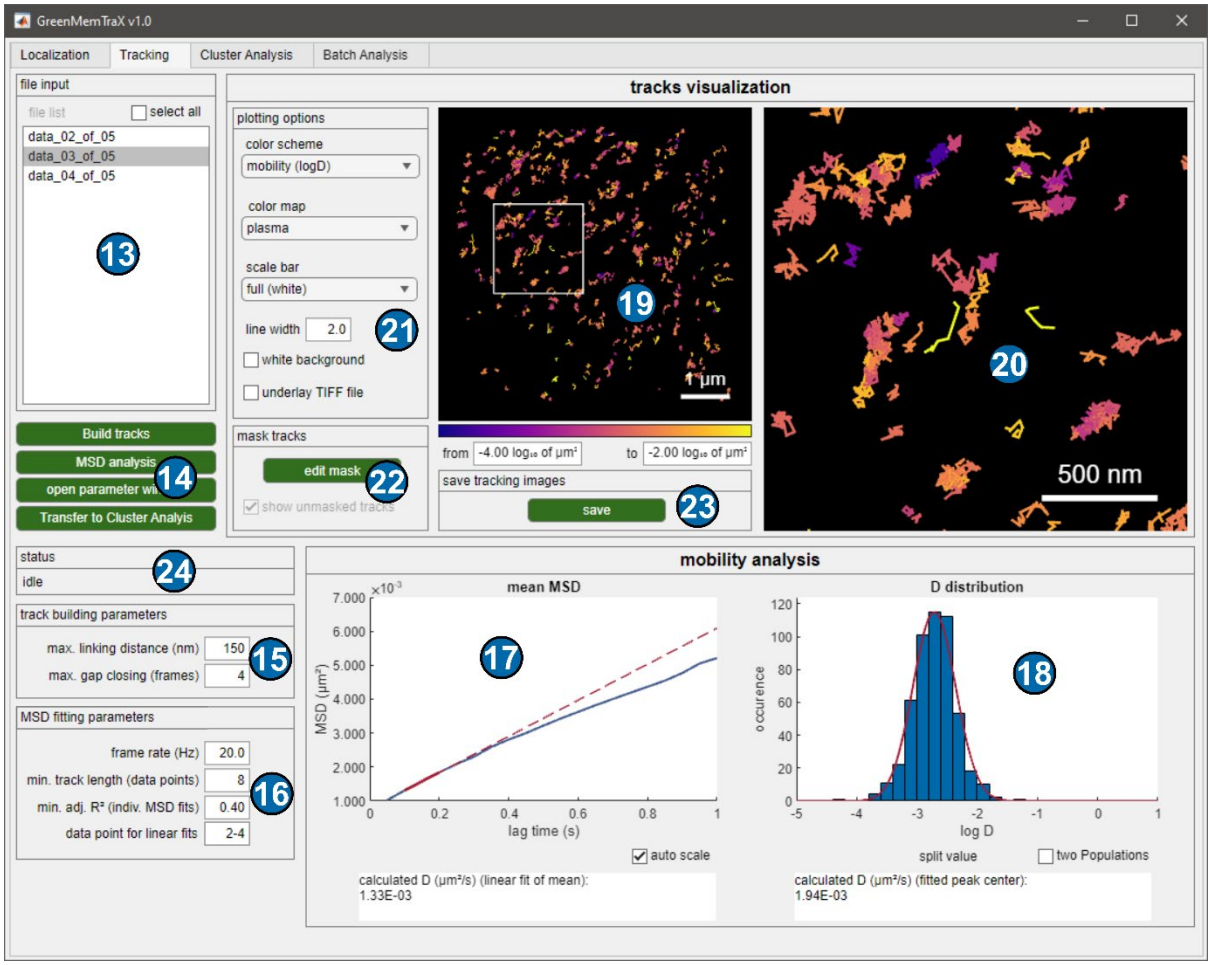

Figure 5. Tracking tab of OneFlowTraX, including track building and mobility analysis.

(13) file input panel; (14) buttons for track building, mobility (MSD) analysis, opening the parameter window and transferring the data to the Cluster Analysis tab; (15) track building parameters, (16) MSD fitting parameters, (17) mean MSD plot, (18) histogram showing the distribution of diffusion coefficients; (19) tracking image (overview); (20) tracking image (user-defined view); (21) plotting options for the tracking images; (22) mask tracks panel; (23) button to save the tracking images; (24) status panel.

4.2.1 Building tracks

The generally low localization density in sptPALM data allows for a simple, but very fast tracking algorithm that can also bridge detection gaps typical of single-molecule microscopy. The corresponding code was adapted from a MATLAB program published by Jean-Yves Tinevez (Tinevez, 2011). The principle of this simple Linear Assignment Problem (Simple LAP) tracker is also part of his software 'TrackMate' (Tinevez et al., 2017). Its parameters can be set in the track building parameters panel (15).

- max. linking distance (nm): maximum distance between two fluorophore positions in two consecutive frames to be assigned to the same track. Suggested values are discussed in section 4.2.2.
- max. gap closing (frames): maximum number of frames to look ahead when a fluorophore position is not found in the subsequent frame. If one is found in the same vicinity (set by the max. linking distance), they will be assigned to the same track. For most cases, allowing 4 gap frames is a good choice.

Pushing the button ‘Build tracks’ button will then perform the tracking for all files in the list (13), while the status panel (24) will update on the current progress. After selecting a single file from that list, the tracks can be inspected in the tracking images (19) and (20).

4.2.2 Interpreting the tracking results

After performing the track building, selecting a single file will display the results in the tracks visualization panel. The left image (19) gives an overview using the maximum dimensions of the input file, while the right image (20) provides an initial, 10x magnified view of its center. You can then move the view in (20) by dragging the image with the LMB and change the magnification by rotating the mouse wheel. The square in (19) indicates which section of the image is currently displayed in (20). These images can be used to inspect the track building results, especially to identify misconnected tracks. Figure 6 shows three cases that depend on the value of the maximum linking distance that is set in track building parameters panel (15).

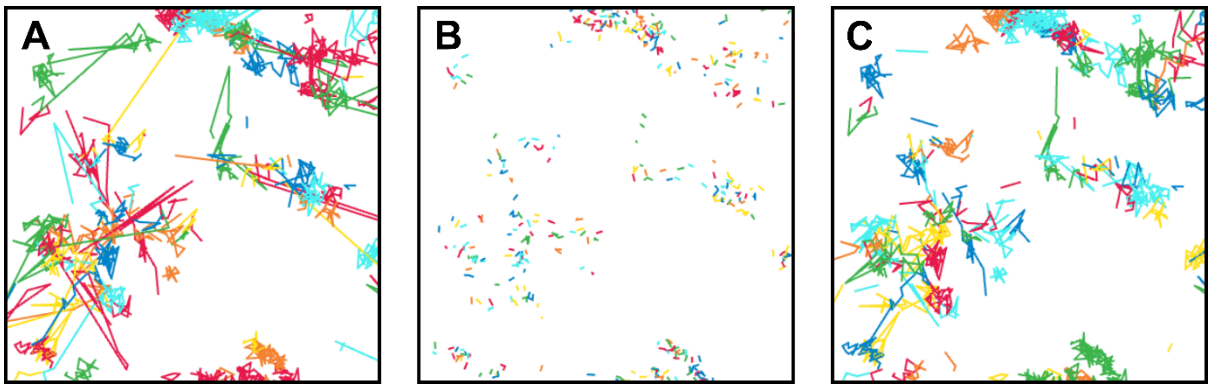

Figure 6. Choosing unsuitable parameters can produce artifacts during track assembly.

A) If the maximum linking distance is set to a value that is too high, distant fluorophore positions are (mis-)connected, leading to long lines in the image that would suggest sudden ‘jumps’ of a membrane protein. B) If the distance value is set too low, tracks can no longer be fully connected, leading to ‘chopped up’ track fragments. C) A suitable maximum linking distance produces appropriately linked tracks. In practice, few (mis-)connections are usually averaged out in the analysis, while too many short track fragments will greatly diminish the available data for mobility analysis.

Such track artifacts especially pose a problem when the fluorophore density per frame was too high, otherwise even large maximum linking distances would not misconnect too many fluorophores. On the other hand, too few fluorophores per frame would require much longer acquisition times. Suitable fluorophore densities depend on the expression level of the protein, the power of the activation laser for photoactivation/-conversion and many other things. It is a matter of experience to find the right conditions for the samples and the data acquisition, with some protocols suggesting suitable settings (Bayle et al., 2021). In our experiments, recording slower receptor proteins, 150-200 nm for the maximum linking distance was usually a good starting point. By visualizing the track results in (19) and (20), you can easily inspect the quality of the track building and either adjust the track building parameters (15) or fine-tune the sample preparation and image acquisition to enhance the quality of the raw sptPALM data.

4.2.3 Mobility analysis

One of the most common methods to analyze membrane protein dynamics for sptPALM data is done via mean square displacement (MSD) curves, indicating how far a particle under Brownian motion has traveled over a time interval τ (also called the lag time). For experimental data that will usually feature tracks of very different lengths (see Figure 7A), the following formulation of MSD can be applied to each track (Manzo and Garcia-Parajo, 2015):

$$MSD(\tau = m\Delta t) = \frac{1}{N-m}\sum_{i=1}^{N-m}\left[\boldsymbol{x}_j(t_i + m\Delta t) - \boldsymbol{x}_j(t_i)\right]^2, \qquad m = 1,2, \ldots N-1$$

where N is the total number of localized fluorophores in track *j* and $\boldsymbol{x}_i$ is the *i*-th position (in 2D) within the track. In sptPALM movies, the smallest measurable time interval Δt is the time between two frames (also compare Figure 7B).

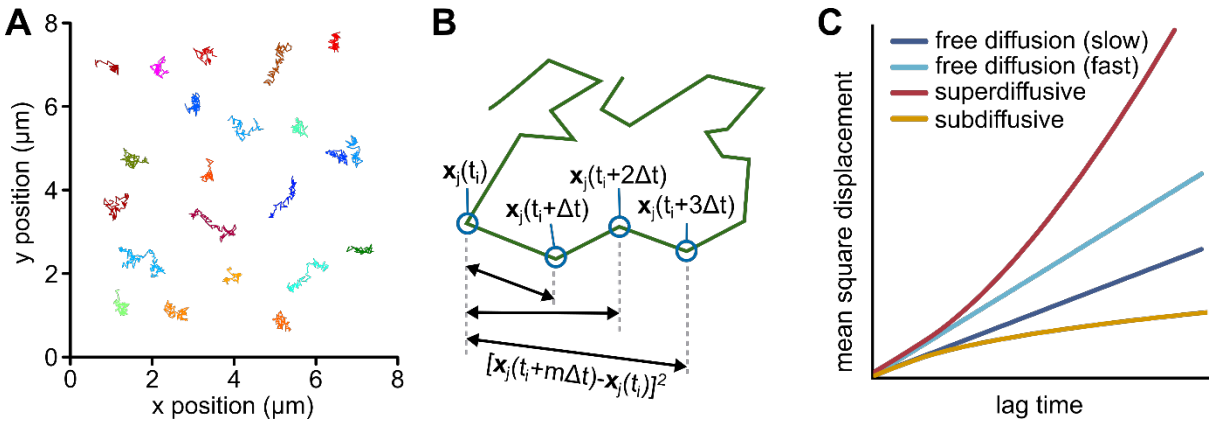

Figure 7. Track analysis and mean square displacement (MSD) plots.

A) Example of single molecule tracks. B) Calculation of MSD values for a single track (green). C) Exemplary MSD plots of Brownian movement (free diffusion) with a low (dark blue) or high (light blue) diffusion coefficient, superdiffusive behavior (red) and subdiffusive behavior (yellow). Adapted from (Manzo and Garcia-Parajo, 2015).

Accordingly, the first lag time (m = 1) in an MSD curve is $\tau = \Delta t$, including all averaged squared fluorophore travel distances between two consecutive frames. The next lag time (m=2) is $\tau = 2\Delta t$, including all averaged squared fluorophore travel distances between frame 1 and 3, 2 and 4, 3 and 5 and so on. The last lag time $\tau = (N-m)\Delta t$ in the MSD curve shows the squared distance between the first and last point in the track. Obviously, many more distances can be averaged for smaller time intervals, so their MSD values have higher statistical relevance than those for larger intervals. The MSD curve for a particle undergoing Brownian motion in two dimensions under ideal conditions then scales linearly with the lag time (Manzo and Garcia-Parajo, 2015):

$$MSD(\tau) = 4D\tau$$

In consequence, the diffusion coefficient D could be easily obtained by a linear fit of the MSD curve, its slope already indicating slower or faster diffusing species (Figure 7C, dark/light blue). Individual tracks in biological systems like membranes, however, often deviate from this ideal diffusive behavior (Figure 7C, red/yellow). Nevertheless, by selecting tracks with good quality, limiting the lag time range that is fitted and only keeping reasonable fit results (see the parameters below), a reasonable estimation of D is still possible and common practice in the community (Bayle *et al.*, 2021; Hosy et al., 2015; Martiniere et al., 2019). To account for these adjustments, the obtained diffusion coefficients are often referred to as 'apparent' or 'instantaneous' diffusion coefficients. Thus, despite all the simplifications, a characteristic value for protein mobility can be calculated that can be compared between experiments. You can set the above-mentioned parameters in the MSD fitting parameters panel (16).

- frame rate (Hz): frequency (in frames per second) at which consecutive images were acquired. This value usually ranges between 10-100 Hz and largely depends on the signal-to-noise ratio. You should aim for the highest frame rate that still delivers good contrast for the

localization algorithm to work properly. Note that only experiments using the same frame rate can strictly be compared to each other.

- min. track length: minimum track length (in frames, including gaps) to qualify for MSD analysis. With a typical value of 8 frames, this will discard up to half the recorded tracks.
- min. adj. R^2: minimum adjusted R^2 value for the linear fit to qualify its results (i.e., the diffusion coefficient) for subsequent analyses. A value of 0.4 reliably discards unsound fitting results without losing too much data.
- N points for linear fits: Only the first N time lags (typically 4) will be included in the linear fit of individual MSD curves, as they have the highest statistical relevance (see above). Additionally, diffusive behavior that diverges from Brownian motion usually only becomes noticeable at higher time lags. If a track's MSD curve misses any of these points, it will be discarded entirely. You can also enter an interval, e.g., '2-4', to only include the data points 2, 3 and 4 of the MSD curve in the linear fit. This is often done as the first lag time shows 'complex behavior' (Kusumi et al., 1993).

After checking/setting these parameters you can start the analysis by pushing the button 'MSD analysis' (14) that has turned green as soon as tracking was performed for the files in the list (13). The results of the MSD analysis are shown in the plots (17) and (18).

4.2.4 Interpreting the mobility analysis results

After performing the MSD analysis, the results will be plotted in (17) and (18). As with the localization results, data is accumulated and plotted for all currently selected files, so you must select a single file to see its individual results. The left plot (17) represents the mean (time-ensemble-averaged) MSD that averages the corresponding lag times of all tracks in the selected file(s). The default lag time scale is set to one second, but the auto scale box below can be unchecked to view the entire graph. As mentioned above, the MSD values of larger lag times are dominated by the longest (and thus, also fewest number of) tracks, so noisy deviations from the MSD curve types shown in Figure 7C in that region are usually no reason for concern. For the calculation of the diffusion coefficients, only the linear fit of the first few points is of interest, marked as a solid red line (the dashed red line is an extension of the linear fit as a guide for the eye). However, you should take note if even this fitting region is very noisy (indicating very few localized fluorophores) or whether the curve is overall steeply parabolic (which might indicate sample or microscope stage movement) and re-check the raw sptPALM data accordingly. The result of the linear fit is shown below the graph as the calculated diffusion coefficient, but this ensemble value is only seldom used in further analyses. In the batch analysis (see section 4.4), the mean MSD plots including their error bars (not shown in (17)) are stored in the results file.

On the right side (18), the MSD curve of each track of the selected file(s) was fitted to produce a list of individual diffusion coefficients that are plotted here as a histogram, with their logarithmized values typically showing a bell-shaped distribution (Figure 8A). In the case of only one discernable diffusing molecular species in the sample, this distribution can be fitted with a Gaussian function to calculate the average diffusion coefficient of all tracks in the selected file(s). In practice, this is the value (repeated for many sptPALM movies, i.e., single cells) that is used in many published boxplots to show mobility differences in comparative experiments. The advantage of these distribution plots is that they can also uncover subpopulations hidden in the mean MSD plots (Figure 8B, C).

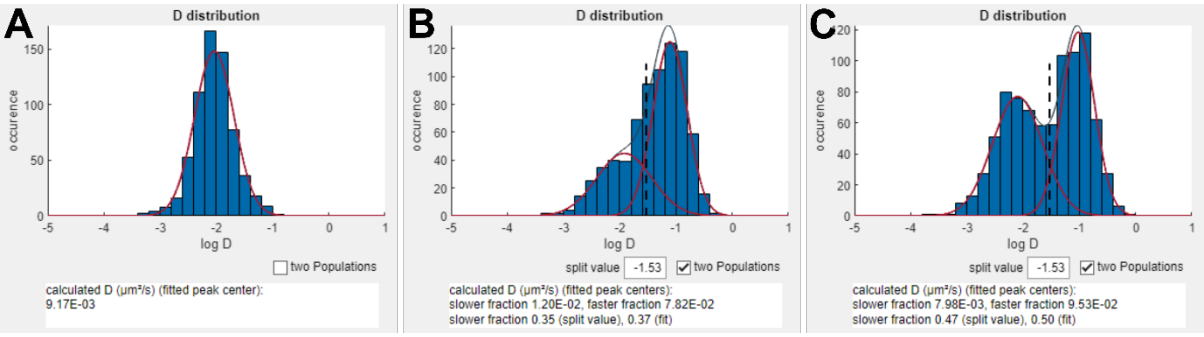

Figure 8. Exemplary distributions of diffusion coefficients.

A) Case with one diffusing species. The histogram (blue bars) can be fitted with a Gaussian function (solid red line) to calculate the average diffusion coefficient as its peak center. B) Case with two differently diffusing species. Checking the 'two populations' box, the histogram is fitted with two Gaussian functions to calculate the diffusion coefficients of both populations. The fraction of the slower population is either calculated by a user-defined threshold (split value, also visible as the vertical dashed line) or via the ratio of the Gaussian fit functions. C) Like B), but with a higher fraction of the slower population.

When two populations are clearly visible, you can check the 'two populations' box. The program will attempt to fit the histogram data with a two-component Gaussian mixture model to estimate each population's diffusion coefficient. In addition, the fraction of the slower population is obtained from this fit. Alternatively, you can enter a threshold or split value that is also visible as a vertical dashed line in Figure 8B, C., The slower fraction is then given as the fraction of individual diffusion coefficients left of that line (i.e., slower than the split value) in relation to their total number. The fitting procedures implemented here are one of many ways to analyze this data. We acknowledge that many researchers have their own processes for statistical population analysis, so the batch analysis (see section 4.4.3) will also deliver the lists of individual diffusion coefficients as the starting point for their own statistical analyses.

4.2.5 Track visualization options and saving of images

The plotting options panel (21) offers several styles to adjust the tracking images.

- color scheme: determines which parameter is used to color the tracks. Most options use a continuous color map (see below) that is displayed as a color bar gradient under (19). Below that bar, you can set the range of values in the sample that are mapped to that gradient. Tooltips show the corresponding units and allowed values that you can enter into the edit fields.
    - random: each track is colored randomly from a set of 4, 6, 8 or 10 colorblind-friendly colors
    - duration: time span in seconds (first to last detected frame in seconds)
    - displacement: length of the track from the first to last detected frame in nanometers
    - range: each track is encompassed with a box and their diagonals serve as an estimation of the tracks' ranges in nanometers
    - mobility (D): apparent diffusion coefficients in µm²/s
    - mobility (logD): logarithmized apparent diffusion coefficients
    - immobile/mobile (logD): two colors, determined by the entered split value below the color bar, based on the logarithmized apparent diffusion coefficients
- color map: colors or color gradient that is also visible as color bar under (19).
    - cycle 4/6/8/10 colors (for the 'random' color scheme)
    - inferno, plasma and hot color maps (and their reverse) for continuous gradients

    - white and black for monochrome coloring
    - orange/blue, red/teal (and their reverse) as binary color options for the 'immobile/mobile (logD)' color scheme
- scale bar: puts scale bars in both images. The scale bar in (20) is updated as soon as you finish adjusting the view. You can choose between (white/black in parentheses determines the color):
    - none: no scale bar
    - bar: a rectangular, horizontal scale bar
    - full: the bar with its length shown above
- line width: width in points (1 point = 1/72 of an inch) of the track lines in (20). The tracks in (19) will be displayed as 1/3 of this value.
- white background: can be checked for a white background. The square in (19) will be painted in the contrasting color
- underlay TIFF file: when checked, the image background changes to a maximum intensity projection (MIP) of the input sptPALM movie.

The currently displayed images can be saved as a vector graphics (.pdf) file by pushing the 'save' button (23). It will be stored in the same folder as the corresponding input data file, using the same name, but with the added suffix '_tracksVis' (with the left image as the first, the right image as the second, and the colorbar (including units) as the third page of the .pdf file).

4.2.6 Masking of tracks

Sometimes the recorded area in sptPALM movies is larger than the extent of the observed cell. However, especially at the visible cell boundary, the curvature of the membrane can strongly influence the calculated MSD values. Ideally, only the tracks in the more planar regions of the membrane should be used for further analysis. For this purpose, we have developed a small program to mask areas of the sptPALM movies. If you click on the 'edit mask' button in the mask tracks panel (22), the sptPALM mask editor will be launched in a new window. When no mask file was previously defined (see below if otherwise), the GUI will present as in Figure 9A.

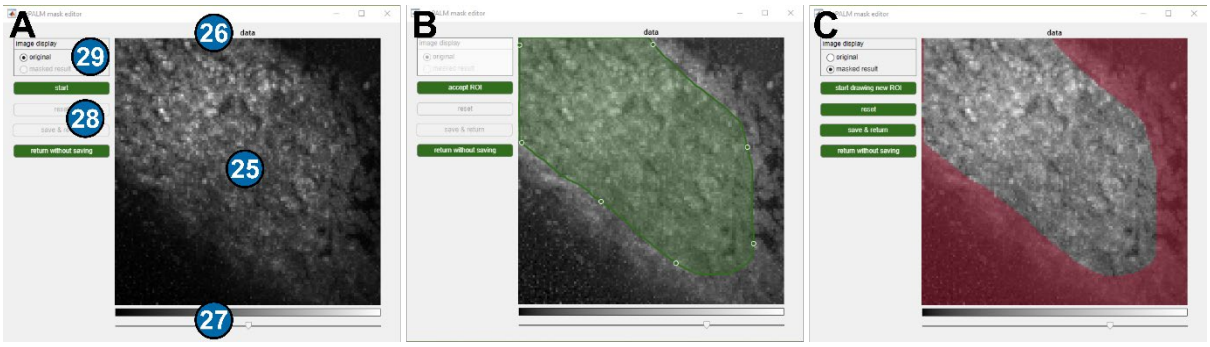

Figure 9. Graphical user interface of the mask editor.

A) Appearance of the interface when it is first started from OneFlowTraX, with (25) display of the maximum intensity projection (MIP) of the sptPALM input file; (26) name of the input file; (27) contrast slider for the MIP; (28) buttons to start drawing/accept/redraw a range of interest (ROI) as mask, reset to the initial state, save the mask and return to OneFlowTraX, return without saving the mask; (29) switch between the original and masked view. B) User-drawn mask, shown in green with darker green boundary and adjustment points. C) Visualization of how the mask will be applied to the data (the red-tinted regions will be disregarded during subsequent analyses).

The maximum intensity projection (MIP) of the input sptPALM movie is shown in (25) with its name shown above (26). A contrast slider (27) can be used to adjust the brightness of the MIP at any time.

By pushing the ‘start drawing new ROI’ button, the mouse pointer changes to a cursor in (25) and a ROI can be drawn by pressing down the LMB and tracing the contour of the cell. The ROI will auto-close when the LMB is released and look like in Figure 9B. The ROI border features several points that can be dragged for adjustments, more points can also be added by double-clicking, and the entire ROI can be moved by dragging it to a new position. When you are satisfied with the ROI, you can click ‘accept ROI’ (the uppermost button changed its text), if not, still click ‘accept ROI’ first, then redraw the ROI by once again clicking the ‘start drawing new ROI’ button. When accepted, the resulting mask will present as in Figure 9C, showing the regions in red that will be disregarded in subsequent analyses. You can now switch back and forth between the original and masked view using the radio buttons (29). Pushing the ‘return without saving’ button or closing the window will return to OneFlowTraX without any changes. If you choose to save the mask by clicking ‘save & return’, the mask will be saved as a .TIF file in the same folder as the original input file, with the same name, but with the added suffix ‘_mask’. This file could also be edited outside OneFlowTraX, has the same x- and y-dimensions as the original input file, with white pixels for areas that will be retained and black pixels for areas that will be disregarded.

If a mask file was already present when starting the sptPALM mask editor, it will be automatically loaded into the mask editor (looking like Figure 9C), but you can still redraw this mask and update the stored one if needed. Note that the mask editor can also read other image files that can be thresholded and interpreted as masks (.png, .jpg etc.) if they are in the same folder as the original input file, with the same name, but with the added suffix ‘_mask’. If such a mask file already existed or was now newly created, it will automatically be applied to the Tracking tab of OneFlowTraX by (re-)building the tracks using only those localized positions that are inside the user-defined mask (Figure 10).

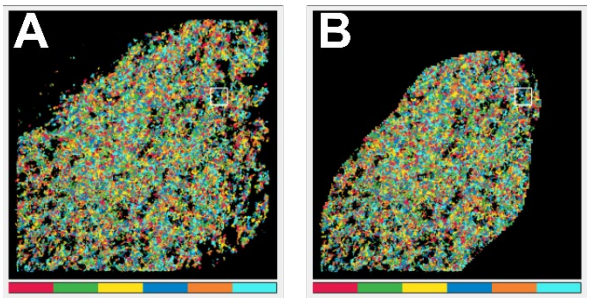

Figure 10. Effect of applying a mask to the track data.

A) Original view. B) Mask in effect, localization data in masked regions was ignored during track building.

You can switch back to the original (unmasked) view by checking the ‘show unmasked tracks’ box in the mask tracks panel (22), which is only enabled when a mask file was found. Note that the existence of a valid mask file will always apply its effect to the data. If you want to remove existing masks, it is best to delete them outside OneFlowTraX.

4.2.7 Transferring the tracks to the Cluster Analysis tab

After track building and the mobility analysis were performed, the ‘Transfer to Cluster Analysis’ button (14) turns green and allows to transfer the data of the currently selected files to the Cluster Analysis tab. The batch analysis (see section 4.4.1) features some additional options for filtering the tracks (minimum number of localizations, duration, displacement and range). Note that, even if you should choose to only use localization data for the cluster analysis, you still need to perform all analysis steps in the Tracking tab to continue as some visualization options of the Cluster Analysis tab

rely on the tracks and also on the MSD results. In the batch analysis (see section 4.4.1), these steps can be omitted. Pushing the transfer button will automatically switch to the Cluster Analysis tab.

4.3 Cluster Analysis tab

This tab (shown in Figure 11) comprises a selection of cluster analysis algorithms that can be applied either to the tracks, their centroids or all localized positions. The first of the transferred files will be selected automatically to display their tracks in the large overview window (31), where you can zoom in and out by rotating the mouse wheel and move around by dragging the image with the LMB.

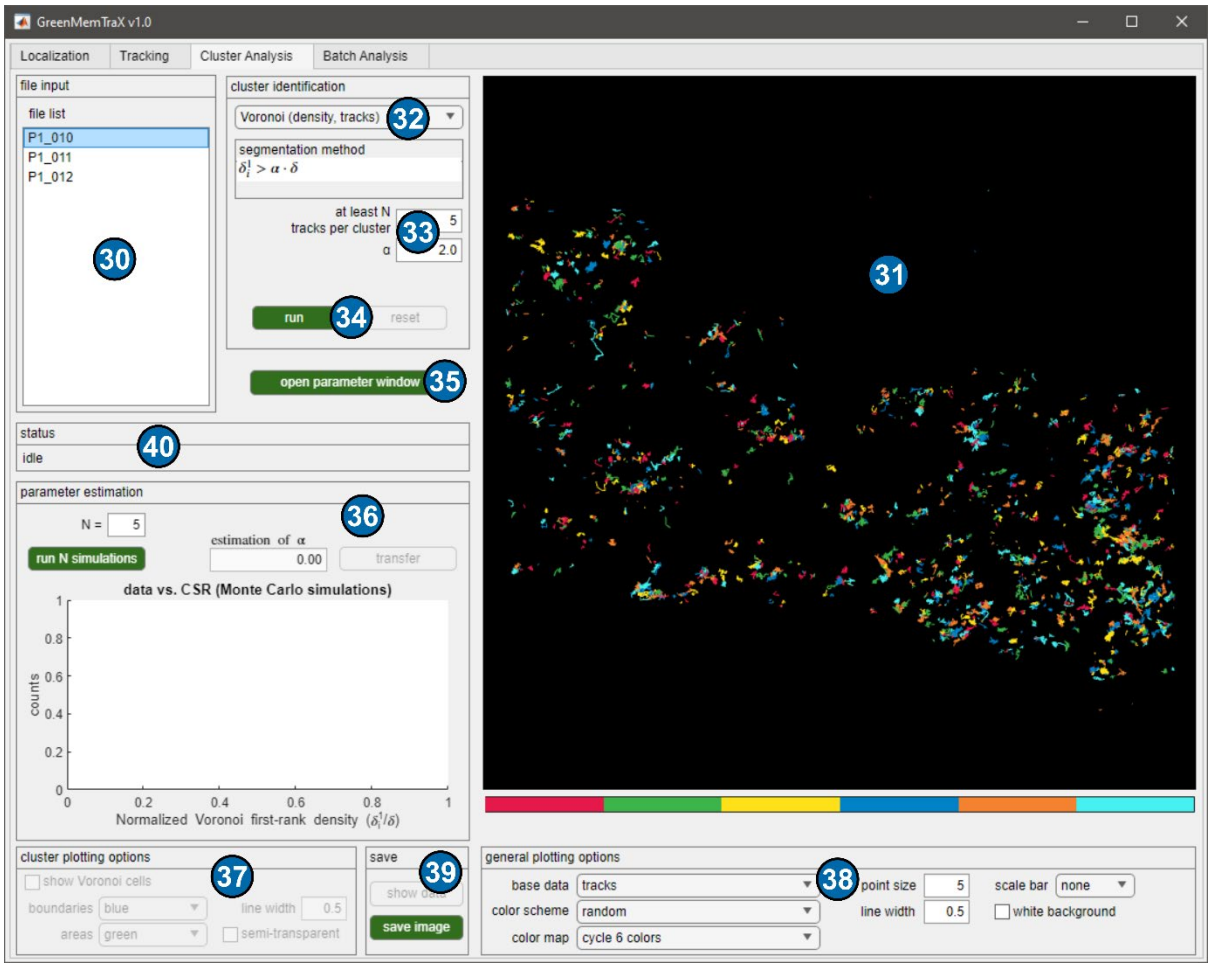

Figure 11. Cluster Analysis tab of OneFlowTraX, implementing several clustering algorithms and their visualizations.

(30) file input panel; (31) data visualization (user-defined view); (32) cluster algorithm drop-down menu; (33) parameters for the chosen cluster algorithm; (34) buttons to run the chosen cluster analysis or reset their options; (35) button to open the parameter window; (36) parameter estimation panel; (37) plotting options for clusters; (38) general plotting options; (39) save panel; (40) status panel.

4.3.1 General remarks for cluster analyses of sptPALM data

While the algorithms presented here can robustly reproduce the ground truth of idealized simulated data, it is much more difficult to provide absolute information (e.g., size, shape, and number of proteins in clusters) in real biological systems. Based on current knowledge, it is even difficult to define a cluster as such because, depending on the protein, it is not known whether they have hard or soft boundaries, are in constant exchange with other membrane proteins, or what distinguishes functional units from random agglomerations.

One of the most promising aspects of using sptPALM data for cluster analysis is the correlation of tracking and mobility data of individual membrane proteins with their organization in clusters. However, this also involves some disadvantages (compared with fixed, immunolabeled samples and structure resolving SMLM techniques like PALM or STORM):

- Individual tracks must be distinguishable, so the density of visible fluorophores must be much smaller.
- The acquisition time is limited by the longevity of the observed cell and by the bleaching of the typically used FPs.
- Detected fluorophores that belong to the same membrane protein (tracks) should be reduced to one position (the track centroid), further decreasing the amount of available positional information for cluster analysis.

The alternative would be to work with much higher fluorophore densities and laser intensities to determine the position of as many individual membrane proteins as possible. However, this would be incompatible with tracking, thus preventing correlations of tracking with the cluster data. High intensity laser illumination is also a major problem for living cells, especially in plants.

Although Voronoi tessellation for cluster analysis has recently become popular in plant research (mainly based on the SR-Tesseler software), the analyses are still subject to some uncertainties. For example, it is easy to show that the cluster-specific parameters need to be changed only slightly to significantly alter calculated cluster sizes. Moreover, there is no common practice yet on which data the cluster analysis should be based on - the track centroids or all localized fluorophore positions.

Nevertheless, we believe that cluster analysis of sptPALM data is a very important tool for studying membrane protein organization. Although the algorithms are limited in their ability to reliably calculate absolute values, comparative analyses, e.g., increasing/decreasing clusters sizes for different treatments or genotypes, are quite acceptable. This is even more true when different algorithms (with their parameters fixed over a series of experiments) show the same trends. For this reason, we recommend always trying several cluster algorithms to cross-check the results. OneFlowTraX offers several commonly used clustering algorithms and lets the user choose to use the track centroids or all localized positions.

4.3.2 Overview of implemented cluster algorithms

In the following, the algorithms are shortly described, while specific information can be found in the corresponding original publications.

4.3.2.1 Voronoi tessellation

A Voronoi diagram assigns each point in a set to a corresponding region of influence that depends on its distance to other points in the set. Each point can now be represented by its Voronoi cell for which (in 2D) the area is calculated. Smaller areas correlate with a higher density of points in that region, so clusters can be assigned by choosing an area threshold value and grouping all directly touching Voronoi cells (with areas smaller than the chosen threshold value) into clusters. This area-based Voronoi tessellation is used in ClusterViSu (Andronov et al., 2016) that also provides an estimation of the threshold value (see section 4.3.4). The area for each cluster is calculated by summing up the area of all Voronoi cells in that cluster.

A different approach in realized in SR-Tesseler (Levet et al., 2015) that uses the density (i.e., the reciprocal value of the area), but considers all directly neighboring Voronoi cells to calculate a so called first-rank density value for the center Voronoi cell. The threshold value α is specified here as a multiple of the average density (the total number of points divided by the total area), and all Voronoi

cells that satisfy this minimum density value and border each other are then grouped into clusters. Here, the area of each cluster is defined by the area of the convex hull that encompasses all points in that cluster. A custom estimation of α based on the input data is detailed in section 4.3.4.

4.3.2.2 Density-based spatial clustering of applications with noise (DBSCAN)

This simple but very efficient algorithm (Ester et al., 1996) is governed by two parameters, the neighborhood radius (ϵ) that draws a circle around a point and the minimum number of points (minPts) that must be found inside this circle. This is checked for each point in the sample and determines if it is designated as a 'core' cluster point or 'noise'. Clusters are then assigned by looking which points can 'reach' each other, meaning that they are either directly within ϵ of a core point or can be connected via a path of reachable points (border points). To start with, ϵ can be estimated using a k-distance graph (see section 4.3.4). Since cluster sizes are not a direct output of this algorithm, we choose to calculate them in the same way as the density-based Voronoi tessellation (see above), using the area of the convex hull encompassing all points in a cluster.

4.3.2.3 Nanoscale spatiotemporal indexing clustering (NASTIC)

In contrast to the other algorithms, this very recent approach by (Wallis et al., 2023) has the advantage that it is specifically designed for track data and clusters those tracks that spatially overlap (it also allows the investigation of temporal overlap, but this is not yet implemented in this version of OneFlowTraX). A circle is created for each track that has the same area as the track's convex hull and is centered on the track's centroid position. The circle is then fitted in a square (also called an axis-aligned bounding box, AABB) that represents the track's spatial extent. To better approximate elliptical track shapes, each circle radius is multiplied by a global radius factor. Overlapping AABBs can be quickly found and grouped into clusters. The size of each cluster is given by the convex hull that encompasses all localized positions of all tracks in that cluster.

4.3.3 Running a cluster algorithm

You can choose between the different cluster algorithms using the drop-down menu (32). All algorithms have specific parameters (33), but they all feature the 'at least N tracks per cluster' / 'at least N localizations per cluster' edit field, where you can enter the minimum number of tracks / localizations that must be part of the cluster. Potential clusters that do not fulfill this condition are discarded (e.g., to limit accidental cluster assignments or background artifacts).

- Voronoi (density, tracks): density-based Voronoi tessellation, applied to track centroids. The α value sets how many times higher the first-rank density of a Voronoi cell must be than the mean total density to be considered as part of a potential cluster.
- Voronoi (area, tracks): area-based Voronoi tessellation, applied to track centroids. The area of a Voronoi cell must be smaller than the set $area_{thresh}$ value to be considered as part of a potential cluster.
- Voronoi (density, localizations): as Voronoi (density, tracks) but applied to all fluorophore localizations.
- Voronoi (area, localizations): as Voronoi (area, tracks) but applied to all fluorophore localizations.
- DBSCAN (tracks): DBSCAN algorithm, applied to track centroids. Additional parameters set the minimum number of points that must be found in the neighborhood radius ϵ (in nm).
- DBSCAN (localizations): as DBSCAN (tracks) but applied to all fluorophore localizations.
- NASTIC (tracks): NASTIC algorithm, applied to tracks. In our hands, good choices for the radius factor range between 1.0 and 1.3.

The cluster analysis with the chosen algorithm and parameters can then be started for the selected file by pushing the ‘run’ button (34), with the status panel (40) informing on the current progress.

4.3.4 Estimation of cluster-specific parameters

For appropriate initial values, parameter estimations are possible for some of the implemented cluster algorithms, which can be performed in the parameter estimation panel (36).

- $area_{thres}$ for area-based Voronoi tessellation
  As described in (Andronov *et al.*, 2016), an estimation for $area_{thresh}$ can be performed by running Monte-Carlo simulations using the same dimensions and number of points as the input data, but with all points randomly placed (complete spatial randomness, CSR). You can choose a value for the number of simulations (due to the computational load it is best to start with low value like 5) and push the ‘run N simulations’ button. The number of pending simulations is updated in the status panel (40). Afterwards, the distribution of Voronoi cell areas of the input data (blue) is compared with the averaged distribution of the CSR simulations (black), visualized in the plot in (36). After the first crossing of these histograms, the Voronoi cell areas are predominantly associated with CSR. This calculated threshold value is displayed above the plot and can update the $area_{thresh}$ value (33) by pushing the ‘transfer’ button.

- α for density-based Voronoi tessellation
  The corresponding publication (Levet *et al.*, 2015) offers no method to estimate of the α value, so we chose to implement a method analogous to the $area_{thresh}$ estimation (see above). The plot will now show the comparison of input data (blue) with simulated CSR data (black), based on the normalized Voronoi first-rank density. The estimation of α is shown above the plot can update the corresponding value (33) by pushing the ‘transfer’ button.

- ϵ for DBSCAN
  The minimum points parameter value of DBSCAN for 2D systems like membranes is 4 but is often set to larger values for noisy data. By pushing the ‘run estimation’ button, the distance of each point in the sample to its k-th nearest neighbor (k being the minimum points parameter value) can be calculated and a plot is generated by simply sorting these distances. The so-called ‘knee’ of this graph is a good estimate for the transition from clustered points (blue area) to noise points (orange area) and is shown above the plot. This estimate for ϵ can then update the corresponding value (33) by pushing the ‘transfer’ button. We suggest first adjusting the minimum points value (if necessary) and then estimating ϵ as a starting point to fine-tune the cluster analysis.

When the estimation fails using the above methods, the ‘transfer’ button is disabled. You can then still try to estimate an appropriate value by visually inspecting the plots, using the estimation guidelines above.

4.3.5 Interpreting the cluster analysis results

After running the chosen cluster analysis algorithm, the results are visualized in (31) and a results table is opened when the ‘show data’ button (39) is pushed. When inspecting (31), you should neither end up with no clusters at all nor with one large cluster that covers all the data. In case this happens due to unsuitable parameters, you can change them and repeat the cluster analysis by pressing the 'run' button again. By choosing different visualization options (see below) you can also check if the calculated cluster boundaries are realistic when compared to the underlying data.

Additional information can be gathered from the results table that lists the calculated data for each cluster with the first row showing the averaged values. The columns of that table include:

- size (nm²): cluster sizes in square nanometers, as calculated by the chosen algorithm (see above).
- diameter (nm): diameters of the clusters in nanometers. For the area-based Voronoi tessellation, the cluster diameter is defined as the diameter *d* of a circle with the same area *A* as the cluster (Andronov *et al.*, 2016) and will also be calculated in OneFlowTraX in that way. For all other cluster algorithms, we chose to calculate the diameter with a method that is closest to the one chosen by (Levet *et al.*, 2015). All localizations/track centroids in a cluster are treated as a point cloud that is fitted by a two-dimensional Gaussian function. Its full width at half maximum (FWHM) values for x and y are then averaged to provide one value that is defined as the cluster's diameter. Should you prefer the diameter calculation of (Andronov *et al.*, 2016), you can easily calculate it from the stored area results after batch analysis (see section 4.4.3) with $d = \sqrt{4A/\pi}$.
- N: number of localizations/tracks in the cluster
- x position (nm): x position of the cluster's centroid in nanometers
- y position (nm): y position of the cluster's centroid in nanometers

4.3.6 Cluster visualization options and saving of images

Even without calculated clusters, the general plotting options panel (38) offers several styles to adjust the visualization of localizations, track centroids and tracks.

- base data: sets the base data of the selected file for plotting, independent of the chosen clustering algorithm.
    - localizations: each detected fluorophore position is plotted as one point (track masks do not apply in this case).
    - track centroids: each track is represented by its centroid (the average of all localized fluorophore positions in the track).
    - tracks: the tracks are visualized as previously shown in the Tracking tab (see section 4.2.5).
- color scheme: determines which parameter is used to color the base data. Many options for tracks use a continuous color map (see below) that is displayed as a color bar gradient under (31). Below that bar, you can set the range of values in the sample that are mapped to that gradient. Tooltips show the corresponding units and acceptable values that you can enter into the edit fields.
    - random: only available for tracks, with each track being colored randomly from a set of 4, 6, 8 or 10 colorblind-friendly colors
    - unicolored: For localizations, this is the only option, showing each localization as a colored point. Track centroids can also be unicolored but can also be colored according to the options below ('track' then refers to their track centroids).
    - duration: time span in seconds (first to last detected frame in seconds)
    - displacement: length of the track from the first to last detected frame in nanometers
    - range: each track is encompassed with a box and their diagonals serve as an estimation of the tracks' ranges in nanometers
    - mobility (D): apparent diffusion coefficients in µm²/s
    - mobility (logD): logarithmized apparent diffusion coefficients
    - immobile/mobile (logD): two colors, determined by the entered split value below the color bar, based on the logarithmized apparent diffusion coefficients

- color (map): colors or color gradient that is also visible as color bar under (19).
    - red, green, yellow and other colors for the unicolored color scheme
    - cycle 4/6/8/10 colors (for the 'random' color scheme of tracks)
    - inferno, plasma and hot color maps (and their reverse) for continuous gradients
    - white and black for monochrome coloring
    - orange/blue, red/teal (and their reverse) as binary color options for the 'immobile/mobile (logD)' color scheme
- scale bar: displays a scale bar, which is updated as soon as you finish adjusting the view. You can choose between (white/black in parentheses determines the color):
    - none: no scale bar
    - bar: a rectangular, horizontal scale bar
    - full: the bar with its length shown above
- point size: size (in points, with 1 point = 1/72 of an inch) of individual localizations and track centroids
- line width: width (in points) of displayed track lines
- white background: can be checked for a white background

Additional options are accessible after running a cluster algorithm. The display will then switch to an appropriate visualization that can be further adjusted by the cluster plotting options (37).

- show Voronoi cells / show DBSCAN assignment: with this checkbox, the cluster-specific visualization can be turned on or off, overriding the general plotting options (38).
    - Voronoi tessellation: the line width now refers to the Voronoi cell lines, while the Voronoi cells are now colored according to the chosen type (area or density) and color map.
    - DBSCAN: the new color scheme options enable to color the points according to their cluster affiliation (randomly chosen for each cluster) or according to their individual designation by the DBSCAN algorithm (noise/cluster) and (noise/core/border) with associated color map choices.
- boundaries: red, green, yellow and other colors for the boundary of individual clusters. You can also choose 'hidden' to conceal all cluster boundaries.
- line width: width (in points) of the cluster boundaries.
- areas: red, green, yellow and other colors for the area of individual clusters. You can also choose 'hidden' to conceal all cluster areas.
- semi-transparent: checking this box will make the cluster areas semi-transparent.

You can also 'mix and match' certain styles, e.g., calculating the clusters with DBSCAN (tracks) that will use the track centroids, but afterwards choosing tracks as the base data with the mobility (D) color scheme. Note that the base data for the cluster analysis is always set by the chosen cluster algorithm (32) and the chosen base data of the plotting options (38) has no influence on that analysis.

The currently displayed image can be saved as vector graphics (.pdf) file by pushing the 'save image' button (39). The .pdf file will be stored in the same folder as the corresponding input data file, using the same name, but with the added suffix '_clusterVis', with the image as the first and the colorbar (including units) as the second page of the .pdf file.

4.4 Batch Analysis tab

While the previous tabs are mainly intended to find suitable parameters for the individual analysis steps, the Batch Analysis tab (Figure 12) now allows to apply them with the chosen parameters to a list of files, storing all the parameters and analysis results in a comprehensive results file.

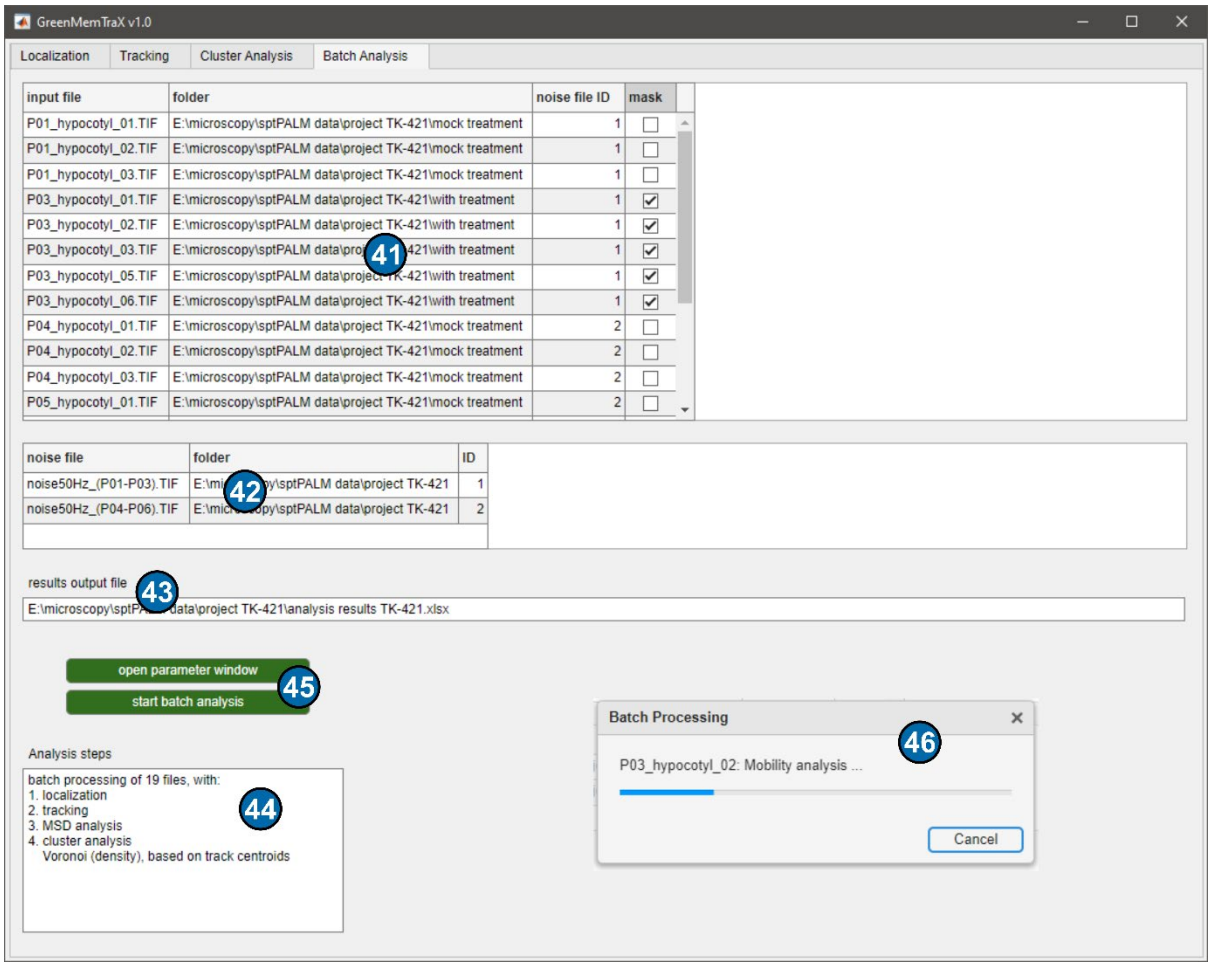

Figure 12. Batch Analysis tab of OneFlowTraX.

(41) input file(s) table; (42) noise file(s) table; (43) output file name and location; (44) overview of the batch processing steps; (45) buttons for opening the parameter window and starting the batch analysis; (46) progress window.

4.4.1 Setting parameters for batch analysis

The parameters you chose during the current session are saved in memory and you can check them in the parameter window by pushing the corresponding button (45), also see section 4.1.8. The input file table (41) is populated by adding files using the parameter window (Files -> Input Files: Add files). You must also supply at least one noise file (Files -> Noise File: Add Files), they will appear in a different table (42) and their assignment to the corresponding input files will be detailed below (see section 4.1.2 when using an EMCCD camera for data acquisition). Additionally, the results file must be specified (Files -> Output File: Choose output file) and will then show up in (43). If any of the above files are missing, the batch analysis cannot start.

You can also define the analysis steps that will be applied (General -> Analysis Steps) by clicking the corresponding check boxes (note that certain analyses depend on each other and will be autoselected). If you choose, you can also output intermediate result lists from the localization and tracking as .dat text files (General -> Raw Data Output) by clicking the corresponding checkboxes and

specifying the file suffixes. All text files are stored in the same folder, with the same name as the corresponding input file, appended with the chosen suffix. The main analysis steps are summarized in ‘Analysis steps’ (44) for review.

For all other parameters, you can find their equivalent in the analysis tabs, except for:

- Tracking -> Limits:
  Only available in batch analysis, this allows you to set additional limits that are applied after building the tracks. The first row, N, is the number of localizations each track consists of. The lower limit is always 3, as this is the minimum number to do subsequent calculations. We provide some other limit options that can be used to filter out tracks that behave noticeably different from the other tracks in the sample. You can inspect if such filtering is necessary by choosing the respective color scheme option in the track visualization (see section 4.2.5). Usually, such artefacts are rare and will not have a big impact on the results, so we recommend leaving these additional limit options at their default values (0 to Inf).

- Cluster Analysis -> Voronoi -> Edge Trimming:
  Localizations cannot be reliably detected near the image borders, so there is always a small strip along each image edge without any localizations. Voronoi diagrams often behave peculiarly at these edges, e.g., producing large Voronoi cells with vertices outside the image frame, complicating the interpretation of their areas. If you unselect ‘trim Edges’, all Voronoi cells with vertices outside the image dimensions will be discarded, which might remove potential clusters near the image borders from analysis.
  When ‘trim Edges’ is checked, you have following options to create a boundary that better fits the existing data (localizations or track centroids):
    - convhull: a convex hull is drawn around the data.
    - AABB (axis-aligned bounding box): a rectangle is drawn that encompasses all data.
    - manual: you can use the now displayed table to enter lower and upper limits for a user-defined boundary rectangle.

  The method then specifies how to deal with Voronoi cells that lie (partly) outside this boundary. With ‘remove’ they will be disregarded entirely, with ‘intersect’, each affected Voronoi cell with be intersected with the boundary to replace it with a trimmed cell. Strictly speaking, these new cells can no longer be called Voronoi cells, but they deliver much more reliable areas and first-rank densities than the original ones at the border regions.

  Note that if an input file uses a mask, these options are overridden, autoselecting ‘trim Edges’, using ‘convhull’ as boundary and ‘intersect’ as method. You cannot directly set these trimming options in the Cluster Analysis tab, but when they are set in the parameter window, you can see their impact when you run a cluster analysis based on Voronoi tessellation in (31). We recommend using the default options (‘trim Edges’ box is checked, ‘AABB’ as boundary with method ‘intersect’) that have produced reliable results in our hands.

When input files are specified, they appear in the file list (41), with their name, their folder, their noise file ID and a checkbox if they use a mask.

The ‘noise file ID’ column is editable, allowing you to assign the corresponding noise file ID that is shown in the noise file list (42) to each file (initially, every file will be assigned to the noise file ID 1). Adding or removing any input files will reset the assignment, so this step should be done after the input file list is complete.

You can also edit the 'mask' column – all boxes are autoselected for which a valid mask file (see section 4.2.6) exists. However, you can uncheck each box if you prefer not to use a mask for that file in the batch analysis. Clicking a box for a file with no existing mask file will have no effect (the box will remain unchecked).

4.4.2 Running the batch analysis

After reviewing the file input (41)/(42) and output (43), the analysis steps (44) and the parameters in the parameter window, the batch processing is started by pushing the 'start batch analysis' (45) button. A progress dialog window (46) will then open, informing about the currently processed file and the estimated remaining time. If you click the 'Cancel' button or try to close the progress dialog window, the batch processing will finish all remaining analysis steps for the current file and then stop (all results, including those for that last file, will still be saved).

4.4.3 Structure of the results file

All parameters and analysis results will be stored in an Excel spreadsheet file you specified in (43), organized in tabs.

- parameters:
  All parameters that were applied to all files during batch analysis are listed here with their values.
- metadata:
  The first column provides the 'file ID' for each file that was processed, with their filename and location shown as 'filename'. These are the file IDs that all other tabs refer to. Additional columns specify the date that each file was acquired, the filename and location of the corresponding noise and mask file (the latter only if that mask file was actually applied), and the sptPALM movie dimensions (width, height and number of frames).
- localization results:
  These are the analysis results that you would find in (5), detailing the original number of detected localizations. The next three columns show the number of localizations that were classified as faulty for the here specified reasons and the remaining localizations after pre-filtering this nonsensical data. Note that the absolute numbers usually do not sum up, as some localizations show multiple fitting errors. The last column shows the remaining localizations after applying the user-defined histogram limits.
- tracking results:
  Here, the original number of tracks are shown that were assembled from the localization data. The next column details the number of tracks that remained after filtering. By default, this step necessarily removes all tracks with only two localizations (as they cannot be used for MSD analysis), but additional specified filter parameters (see 4.4.1) also apply. The last column shows the number of remaining tracks after the mask file (if applicable) was used.
- mean MSD:
  The first row specifies the file ID, the corresponding mean MSD plot data is then organized in column triplets, containing the lag time, the MSD and the standard error. As the mean MSD plot is based on a weighted average (tracks with more localizations are given more weight), the error for each time lag is defined here as the weighted standard deviation divided by the square root of the number of degrees of freedom in the weighted mean. The standard error is indicated as 'SEM*' to reflect this difference to the classic definition of the standard error.
- D lists:
  Each column contains the diffusion coefficients calculated for the tracks in each file, with their file IDs in the first row.

- D fit (1 pop):
  For each file ID (first column), these are the results of a normal distribution analysis applied to the logarithmized diffusion coefficients (calculated from each viable track in the respective file). The second column shows the number of values in this distribution and the third column shows the mean of this distribution. While this calculation works well to find the average diffusion coefficient value for one population, you can apply your own statistical analysis with the data from the tab 'D lists' (that is the basis of the normal distribution analysis used here).
- D fit (2 pop):
  For each file ID (first column), these are the results of a Gaussian mixture model analysis for two components, using the MATLAB function 'fitgmdist'. Be advised that this very specific fitting option should only be applied to situations where two populations in a sample are clearly discernible. The second column shows the number of values in this distribution, while the third and fourth column give the average diffusion coefficients of the slower and faster fraction, respectively. The column 'fit quality' (ok/not converged/failed) shows if problems occurred during distribution fitting. While non-converged fits are often still reasonable, you should handle their results with caution. For more complicated situations or deeper analyses, you should apply your own statistical methods using the data from the tab 'D lists'.
- slower fraction
  The first and second column show the ID and the number of individual diffusion coefficients for each file. With the user-specified split value (the logarithmic diffusion coefficient that serves as a threshold to define the 'slower' and 'faster' fraction), the fraction of diffusion coefficients that are slower than that value is shown in the third column (the faster fraction can be calculated by subtracting each value from 1). The fourth column provides an alternative method based on a two-component Gaussian mixture model (see tab 'D fit (2 pop), also for the limits of this method and the 'fit quality' column) to calculate the fraction of the slower component.
- cluster results:
  The first row specifies the file ID for the cluster results that are summarized in five columns for each file, with one row for each cluster, and columns: area, diameter, N (number of localizations or tracks in that cluster), and the x and y position of each cluster in the sample. For more details concerning the calculation of the cluster areas and diameters see section 4.3.5.

What are the prerequisites for input files?

The input files must be in TIFF format (typically with the endings .tif/.tiff/.TIF/.TIFF), which is the most common output format for cameras used in SMLM experiments. The implemented TIFF reader should be able to cope with most TIFF files, including the OME-TIFF file format. For unreadable TIFF files, you can try to import/export them in ImageJ via Bio-Formats to see if this solves the problem.

5 References

Andronov, L., Orlov, I., Lutz, Y., Vonesch, J.L., and Klaholz, B.P. (2016). ClusterViSu, a method for clustering of protein complexes by Voronoi tessellation in super-resolution microscopy. Sci. Rep. **6**:24084. 10.1038/srep24084.

Bayle, V., Fiche, J.B., Burny, C., Platre, M.P., Nollmann, M., Martiniere, A., and Jaillais, Y. (2021). Single-particle tracking photoactivated localization microscopy of membrane proteins in living plant tissues. Nat. Protoc. **16**:1600-+. 10.1038/s41596-020-00471-4.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press.
Hosy, E., Martiniere, A., Choquet, D., Maurel, C., and Luu, D.T. (2015). Super-resolved and dynamic imaging of membrane proteins in plant cells reveal contrasting kinetic profiles and multiple confinement mechanisms. Mol. Plant **8**:339-342. 10.1016/j.molp.2014.10.006.
Huang, F., Hartwich, T.M.P., Rivera-Molina, F.E., Lin, Y., Duim, W.C., Long, J.J., Uchil, P.D., Myers, J.R., Baird, M.A., Mothes, W., et al. (2013). Video-rate nanoscopy using sCMOS camera-specific single-molecule localization algorithms. Nat. Methods **10**:653-+. 10.1038/Nmeth.2488.
Kusumi, A., Sako, Y., and Yamamoto, M. (1993). Confined Lateral Diffusion of Membrane-Receptors as Studied by Single-Particle Tracking (Nanovid Microscopy) - Effects of Calcium-Induced Differentiation in Cultured Epithelial-Cells. Biophys. J. **65**:2021-2040. 10.1016/S0006-3495(93)81253-0.
Levet, F., Hosy, E., Kechkar, A., Butler, C., Beghin, A., Choquet, D., and Sibarita, J.B. (2015). SR-Tesseler: a method to segment and quantify localization-based super-resolution microscopy data. Nat. Methods **12**:1065-1071. 10.1038/nmeth.3579.
Manzo, C., and Garcia-Parajo, M.F. (2015). A review of progress in single particle tracking: from methods to biophysical insights. Rep. Prog. Phys. **78**10.1088/0034-4885/78/12/124601.
Martiniere, A., Fiche, J.B., Smokvarska, M., Mari, S., Alcon, C., Dumont, X., Hematy, K., Jaillais, Y., Nollmann, M., and Maurel, C. (2019). Osmotic Stress Activates Two Reactive Oxygen Species Pathways with Distinct Effects on Protein Nanodomains and Diffusion. Plant Physiol. **179**:1581-1593. 10.1104/pp.18.01065.
Ries, J. (2020). SMAP: a modular super-resolution microscopy analysis platform for SMLM data. Nat. Methods **17**:870-872. 10.1038/s41592-020-0938-1.
Tinevez, J.-Y. (2011). simpletracker. https://github.com/tinevez/simpletracker.
Tinevez, J.Y., Perry, N., Schindelin, J., Hoopes, G.M., Reynolds, G.D., Laplantine, E., Bednarek, S.Y., Shorte, S.L., and Eliceiri, K.W. (2017). TrackMate: An open and extensible platform for single-particle tracking. Methods **115**:80-90. 10.1016/j.ymeth.2016.09.016.
Wallis, T.P., Jiang, A., Young, K., Hou, H., Kudo, K., McCann, A.J., Durisic, N., Joensuu, M., Oelz, D., Nguyen, H., et al. (2023). Super-resolved trajectory-derived nanoclustering analysis using spatiotemporal indexing. Nat. Commun. **14**:3353. 10.1038/s41467-023-38866-y.