MSP-toolbox 0.2

Manual guide

2021

Contents

The MSP framework contains two elements: MSP-tracker for particle tracking task and MSP-viewer to examine the extracted trajectories.

1 Installation

To run the MSP framework, first you need install the Conda package manager and use command line to install and run the software. You can download the software from github: *github.com/MariiaVision/msp_tracker*.

Mac

1. Install conda for Mac
2. Open the folder which contains the software in Terminal (In finder, right click on code folder and select New Terminal at Folder)
3. In the terminal run **conda env create -f environment_mac.yml**. It should install all the required packages
4. To run the software, in the software folder activate environment with installed packages **conda activate msp** and run **python msptracker.py** (MSP-tracker) or **python mspviewer.py** (MSP-viewer).

Windows

1. Install Anaconda
2. Launch Anaconda Prompt
3. Navigate to the directory with MSP code and create conda environment with: **conda env create -f environment_win.yml**. It should install all the required packages.
4. To run the software, in the software folder activate environment with installed packages **conda activate msp** and run **python msptracker.py** (MSP-tracker) or **python mspviewer.py** (MSP-viewer).

Linux

1. Install conda for Linux
2. In the terminal open the directory with the software and create environment **conda env create -f environment_linux.yml**.
3. To run the software, in the software folder activate environment with installed packages **conda activate msp** and run **python msptracker.py** (MSP-tracker) or **python mspviewer.py** (MSP-viewer).

2 MSP-tracker

The MSP-tracker is an interactive tool for particle tracking. The interface consist of four components (tabs): detection, linking, run tracking and membrane segmentation. Detection and linking tabs allow the user to set tracking parameters and view the results in the provided image viewer. The "run tracking" tab provides overview of the set parameters and runs the tracking algorithms for the complete image sequence. The "membrane segmentation" tab allows the user to set parameters and run membrane segmentation.

The details of the tracking algorithm can found in [*Dmitrieva 2019*]. If you use the tracking software for your research, please cite the paper.

2.1 Data requirement

The data loaded to the MSP-tracker should be a single channel 8 bit tiff image sequence. The data for particle tracking and membrane segmentation should be loaded separately.

2.2 Setting detection parameters

The detection method first produces a number of candidates using the Multiscale Spot Enhancing Filter (MSSEF), then these candidates are pruned using a light CNN network. The detection section is organised according to this process.

The "detection" tab allows the user to load the image sequence ("Select image sequence" button) and provides an image viewer to inspect the original sequence ("original image" option), the pool of candidates ("candidates" option) and the final detections ("detection" option). The user can then set detection parameters with option to save or load parameters setup. To view the candidates and final detections with current parameter setup, the user need to run the algorithm for the current frame by pressing "Run test" button.

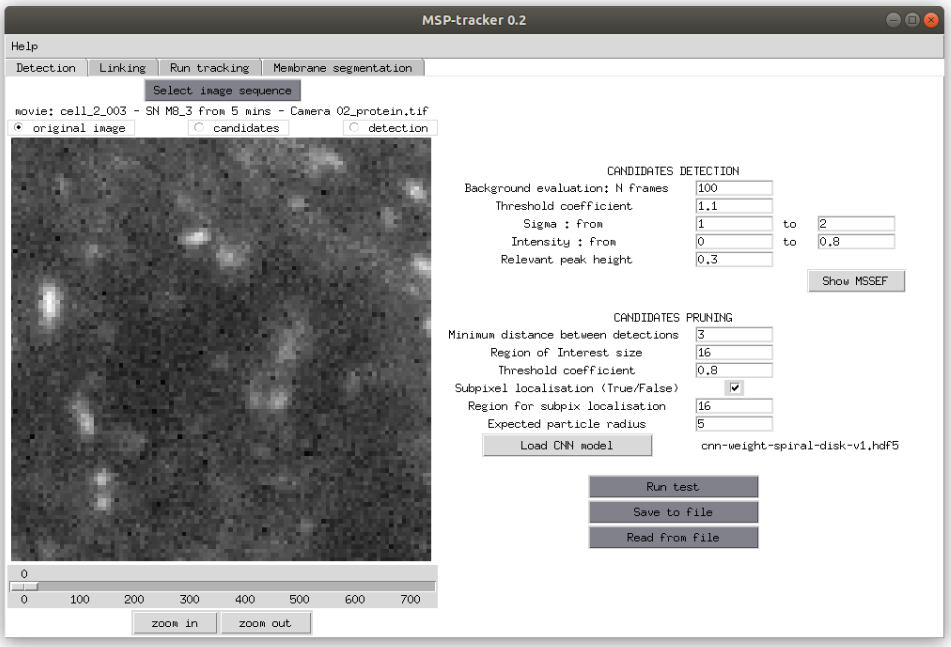
2.2.1 Parameter descriptions

The detection parameters are divided into two groups: candidate detection and candidate pruning. Table 1 provides descriptions of the parameters and their range.

2.2.2 Proposed tuning workflow

1. Set parameters for the candidate detection.
   (a) Set **Background evaluation: N frames**. The parameter is set based on the changes in the background over time. Start using default 100 frames and make changes if necessary. Set a larger number to take into account more frames, too small a value can lead to removing some of the stalled or slow moving particles.
   (b) Adjust Multi Scale Spot Enhancing Filter (MSSEF) choosing **Threshold coefficient** and **Sigma range**. You can use the "Show MSSEF" button to check the enhanced spots. The ideal image shows all the

Figure 1: Detection tab layout



particles of interest as separate spots without any holes inside and the area outside the region of interest doesn't produce any spots.

(c) Adjust intensity range of the detected spots with **Intensity** range values. The value are presented in relation to the intensity of the image sequence: 0 is the minimum intensity value and 1 - maximum intensity value. Set the range to eliminate too bright or too dim spots.

(d) Adjust **Relevant peak height** to include all the particles of interest. Use "candidate" option above the image viewer to monitor the result. Press "Run test" button to run the detection algorithm for the current frame for candidates to show up on the viewer. Increase the value to detect only bright spots and decrease it if darker spots have to be detected as well. Use "Run test" button to update the results after changing the parameter.

2. Set parameters for the candidate pruning.

(a) Load the CNN model which you are planning to use for detection. First try existing options, but if it doesn't perform well, train the model with your data (described in the next Section).

(b) Set Region of Interest based on the chosen CNN model.

(c) Adjust **Threshold coefficient** after running test and looking at the "detection" mode of the image viewer. Decrease the threshold to allow more candidates though the pruning process. Set coefficient to 0 to skip the candidate pruning step.

Parameters	Description	Value range
CANDIDATE DETECTION		
Background evaluation: N frames	Number of frames taken into account for the background subtraction. Start with default 100 frames and make changes if necessary. Set a larger number to take into account more frames, too small a value can lead to removing some of the stalled or slow moving particles.	integer [1, sequence length]
Threshold coefficient	Parameter of the MSSEF which is related to the intensity of the detected spots. Decrease the value to include darker areas, increase to eliminate them.	float [0, 10]
Sigma (from to)	Parameter of the MSSEF which influences size range of the spots. Decrease if you targeting smaller spots, increase to include larger particles. We recommend to start with (1 - 2) and make changes based on the MSSEF image.	float [0.0001:10]
Intensity	Relative intensity range of the detected peaks, where 0 - is the minimum intensity value of the image sequence and 1 - maximum intensity value. Set the range to eliminate too bright or too dim spots.	float [0,1]
Relevant peak height	Proportion of the detected peak in relation to the surrounding environment. Increase the value to detect only brighter/sharper particles and decrease it if darker/misty spots have to be detected as well.	float [0,1]
CANDIDATE PRUNING		
Minimum distance between detections	Minimum number of pixels expected between centers of two particles. If you have dense populated area, decrease the value, larger value allows to eliminate noisy close located detections.	float
Region of image size	Region of Interest for CNN classifier. The number depends on the loaded model (image size it was trained on).	integer (8, 16, 32)
Threshold coefficient	Threshold for the classifier. Decrease the threshold to allow more candidates though the pruning process.	float [0, 1]
Subpixel localisation	on/off subpixel localisation mode. When the mode is off the coordinates of the detected particles are integers, with mode on, the coordinates are refined to provide subpixel location.	True/False
Region for subpixel localisation	Region of interest for the task of subpixel localisation. The value related to the expected particle size and should be adjusted the way that complete particle can fit in. The value should be larger than expected diameter of the particle.	int
Expected particle radius	Expected size of the particle. Adjust the value based on your particle size. Measured in pixels.	odd integer
Load CNN model	Trained model for candidate pruning.	provide path

Table 1: Detection parameters of the MSP-tracker.

(d) Adjust **Minimum distance between detections**.

(e) Adjust settings for the subpixel localisation. The **Region for sub-**

pixel localisation should be larger than double the **expected particle radius**.

3. Test the results on at least a couple of frames. Adjust parameters if required and run test every time you want to see the updated detection result.

4. Save parameters (optional) for future use.

You can test the detection on the current frame – button "Run test", save the parameters into a file – button "Save to file" and load previously saved parameters "Read from file".

2.2.3 Training CNN model

To train the model with a new data, use bash script **run_train_CNN.sh**.

The following variables should be adjusted:

- NUMBER_FILES - number of image sequences used for the training
- MOVIE_PATH_1 ... MOVIE_PATH_N - paths to the image sequence (should be tiff format, single channel)
- POSITIVE_COORDINATES_PATH_1 ... POSITIVE_COORDINATES_PATH_N - paths to the txt file with coordinates of the positive samples
- NEGATIVE_COORDINATES_PATH_1 ... NEGATIVE_COORDINATES_PATH_N - paths to the txt file with coordinates of the negative samples
- IMAGE_PATH - path where the positive and negative samples can be stored
- MODEL_PATH - path where the new weight for the model will be stored
- ROI_SIZE - size of the region of interest used for training (16 or 32), the value depends on the particle size

Provide all the number of paths required in the line: **python tracking_lib...** to include all the training files. Keep the same order of tiff sequences, positive and negative samples.

Train the model by running in command line **bash run_train_CNN.sh** .

Copy the trained model **cnn-model-best.hdf5** to the folder with the existing weights (dl_weight), rename it and select it as a new CNN model in MSP-tracker detection tab.

Preparing data for training: The training data should contain image sequences paired up with two txt files. The txt files contain coordinates of the positive samples (particles of interest) and negative samples (non-particles) in separate files. One raw represents a single sample with the following order (position, x, y, frame):

To create the txt file, you can use ImageJ multi-point tool with measure tool (ctrl+M) to extract coordinates and copy them into a new txt file. The coordinates in the txt file should be represented in pixels (not in real world

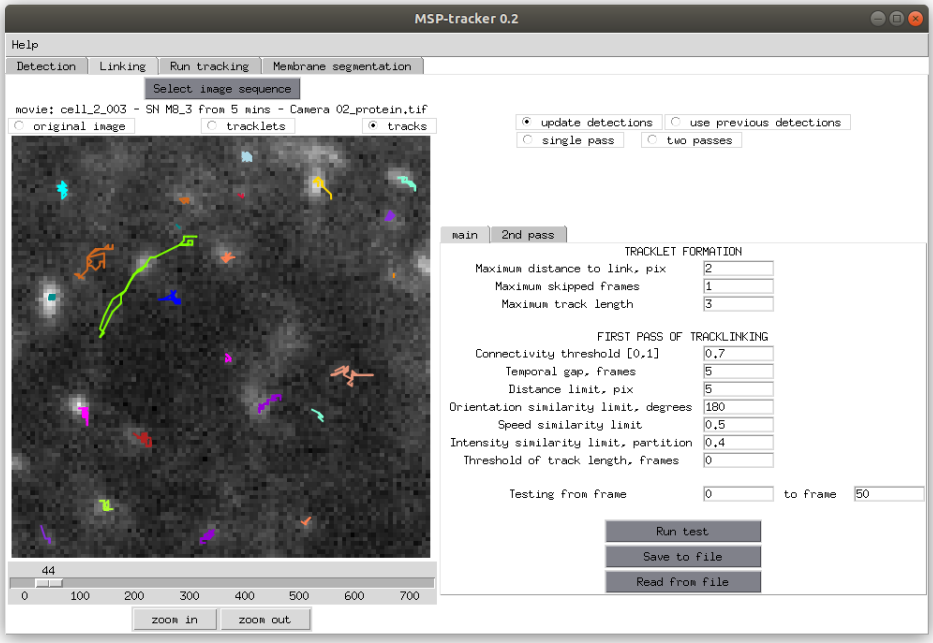
```
42      211.000 338.000 3
43      222.000 314.000 3
44      201.500 300.000 3
45      181.500 273.500 3
46      211.000 72.000  10
47      239.500 84.500  10
48      226.500 84.000  10
49      226.500 219.000 10
```

measurement). It is important to include large variety in the particle class. Non-particle class should include background, bright spots which are not particle of interest, noisy areas without vesicles. Please, select at least 200 samples for each class.

2.3 Setting linking parameters

The linking process is divided into a few steps. Firstly, tracklet formation is performed, where short tracks (tracklets) are formed based on the distance between detections. Secondly, these tracklets are connected between each other (tracklking step). It is possible to use a single tracklinking pass (common solution) or choose to use two passes to extract trajectories for more complex scenarios. Output of tracklet formation and tracklinking can be viewed in image viewer using option "tracklets" and "tracks".

Figure 2: Linking tab layout



2.3.1 Linking parameter description

The linking parameters are divided into two groups - tracklet formation and the tracklining with a choice of "single pass" vs "two passes" for the tracklinking step. When "single pass" is selected only one tab of the parameters should be filled, for the "two passes" option set parameters for the second tracklinking pass in a separate tab. The linking parameters are described in Table 2.

2.3.2 Proposed tuning workflow

1. Choose linking interval ("Testing from frame - to frame"). It can be about 10-40 frames or more depends on the movement you want to capture. The interval should be one of the most challenging in the sequence.

2. Choose between "update detections" and "use previous detections". For the first test run select "update detections" (detection will be run for requested time interval), but when changing parameters for the same frame sequence switch to "use previous detections" (existing detection results will be used). It eliminates the detection process and reduce computation time.

3. Choose the number of passes. This is the number of tracklinking passes. In most cases one pass is enough, but in case of dense vesicle populations or differences in speed movement it can be beneficial to use two passes.

4. Set parameters for the tracklet formation. At this step short tracklets are formed based on the distance. You need to specify three parameters, press button "Run test" and use "tracklets" option above the image viewer to check the formed tracklets. Make sure that the only correct links are created, that means that the links between the points inside each tracklet are correct. Change parameters if required, until the result is satisfying.

5. Set parameters for the first pass of the tracklinking, then if you choose two passes move to the second one. If you have two passes the idea is first to connect slowly moving vesicles:

   - smaller values for speed, intensity and orientation, as it is not important at first pass
   - smaller value for temporal gaps - the connections should be close to each other in time
   - smaller value for the distance limit - this value depends on vesicle movement

   And with second pass faster moving vesicles will be linked:

   - increase values for speed, intensity and orientation, but not necessary
   - increase the temporal gap
   - increase the distance limit

6. Run test and change parameters until satisfied with results and save parameters (optional) for future use.

Parameters	Description	Values
TRACKLET FORMATION		
Maximum distance to link	Number of pixels between detections which still can be linked. Increase the number to connect faster moving vesicles, decrease to create more reliable connections.	float
Maximum skipped frames	Number of frames which can be skipped in the same tracklet between two detections. It is preferable to use small value (0-3 frames) to keep the reliable connections. Increase the value to compensate for persistent failed particle detection.	integer
Maximum track length	Number of frames in one tracklet. Larger value provides faster tracklinking step, but it can cause false linking. We suggest to use 5-10 frames in general, but for a dense movement should be about 3-4.	integer
FIRST/SECOND PASS OF TRACKLINKING		
Connectivity score	Final threshold which decides on tracklets connection. The value is calculated based on the tracklet parameters and shows probability of two tracklets to be connected. Therefore 1 means that only perfectly matching tracklets will be linked, while 0 will allow any sequenced tracklets to be linked. We proposed to use 0.6-0.8 value, increasing it to eliminate unwanted linking.	float [0,1]
Temporal gap	Maximum temporal gap (number of frames) between two connections (when the particle is not detected for a number of frames).	integer
Distance limit	Maximum allowed distance between two detections.	float
Orientation similarity limit	Acceptable difference in orientation.	integer [0, 180]
Speed similairy limit	Acceptable proportional difference in speed.	float [0,1]
Intensity similarity limit	Acceptable difference in intensity.	float [0,1]
Threshold of track length	Limit for the trajectory length. The final tracks shorter than the number will be removed. Should be 0 for first pass if the second pass is used.	integer

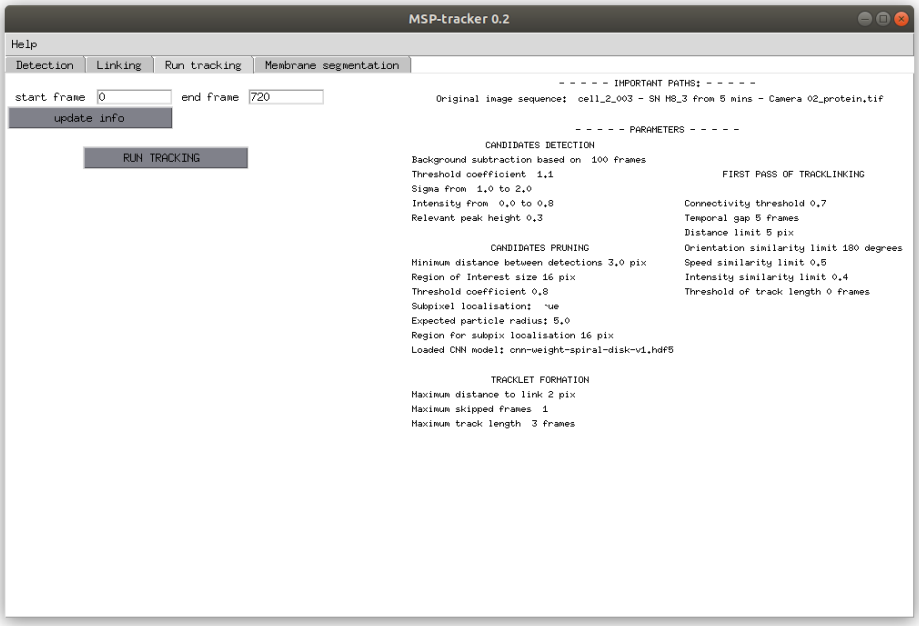
Table 2: Linking parameters of the MSP-tracker.

You can test the detection on the current frame – button “Run test”, save the parameters into a file – button “Save to file” and load previously saved parameters “Read from file”.

2.4 Running the tracker

This tab provides overview of the set parameters, allows the user to set the final result path and run the algorithm for the entire sequence in a provided frame range(set with "start frame" and "end frame"). Use "update info" button to visualise latest updates in parameters and file path names.

Figure 3: Run tracking tab



You can run the MSP-tracker in the following order:

1. Check the parameters.
2. Choose start and end frames
3. Run the tracker with button "RUN TRACKING".

You should not close the software until the tracking is complete. You can follow the progress of the algorithm in the terminal. The processing time depends on the size of the image sequence, set parameters and computational power of the machine it is running on. When finished, the final tracks will appear in the linking window and also can be opened in the separate MSP-viewer software.

2.4.1 Output file format

The extracted trajectory will be saved in two formats: csv and json. In csv file each row represent a single detection with a timestamp and trackID it belongs to, while json file is organised trajectory wise, which each trajectory is described with its ID, coordinates and frames. Both files can be read by the MSP-viewer.

2.4.2 Run the MSP-tracker without GUI

It is possible to run the tracker without GUI when the parameters for the detection and linking are saved to a file. Use **run_msp_tracker.sh** bash script to run the tracker. It is preferable to use the option when the image sequence is large and the tracking can take some time, also it can be beneficial to run the code on server with a higher computational power.

Make sure that the name of the image sequence and parameter files do not have any space inside the name, as it can cause a reading error in the bash script.

Adjust following variables in the bash script **run_msp_tracker.sh** before running the code :

- MOVIE_PATH - path to the image sequence (the file should be tiff format, single channel)
- DETECTION_PARAMETERS_PATH - path to the file with detection parameters (created with the MSP-tracker GUI)
- LINKING_PARAMETERS_PATH - path to the file with linking parameters (created with the MSP-tracker GUI)
- USE_EXIST_DETECTION - "True" or "False": set it to False to run the detection part of the tracker, and True to use existing detections
- DETECTION_PATH - path to the file with the detection for the current image sequence. When USE_EXIST_DETECTION is"False" - detections will be saved there, otherwise (when True) - detections will be read from the file
- RESULT_PATH - path to save the trajectories

Run the script from the command line **bash run_msp_tracker.sh**

2.5 Membrane segmentation

The "membrane segmentation" tab allows the user to set parameters, observe the results in image viewer ("segmentation" option) and run the segmentation on the entire sequence ("Run membrane segmentation" button).The adjusted parameters can be saved to a separate file for future reference.

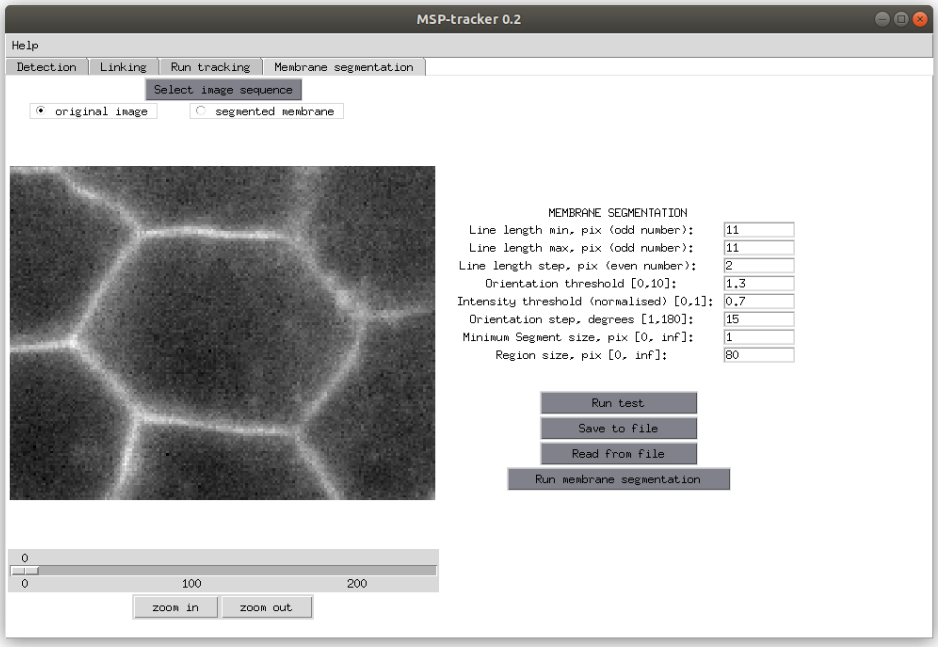
2.5.1 Segmentation parameter

The segmentation approach is based on the multi scale line detector [*Nguyen 2013*] to extract line-like structure of the membrane. The parameters of the segmentation approach and its description are provided in Table 3.

2.5.2 Proposed tuning workflow

1. Set parameters for a single frame. Start with the segment size equal to the frame size. Use "segmentation" option of the image viewer to inspect the results. Start with line settings, orientation and intensity thresholds. Reduce the segment size if the intensity in the frame is not homogeneous.

Figure 4: Membrane segmentation tab



2. Check segmentation for at least couple of frames and adjust parameters if required

3. Save parameters in a separate file (optional)

4. Run the segmentation for the entire sequence (”Run membrane segmentation” button).

You can save the parameters into a file – button “Save to file” and load previously saved parameters “Read from file”.

Parameters	Description	Values
Line length min	The length (in pix) of the shortest line which will be fitted.	even integer
Line length max	The length (in pix) of the longest line which will be fitted.	even integer
Line length step	The step used while iterating over the lines with different length.	odd integer
Orientation threshold	Defines proportion between winning line and other lines.	float [0,5]
Intensity threshold	Defines minimum intensity of the pixel to be considered for the line fitting (normalised value in relation to the maximum intensity in the region)	float [0,1]
Orientation step	Defines a step (in degrees) for line orientation values. Good choice would be number between 10-40 degrees, but the choice should depend on the membrane structure.	integer [1,180]
Minimum Segment size	Defines a requirement to the minimum number of pixels per segment. It allows the user to remove small segments.	integer
Region size	Defines width and height of the region of interest to be processed at once. Use smaller region to compensate for the intensity variation in a single frame.	integer

Table 3: Membrane segmentation parameters.

3 MSP-viewer

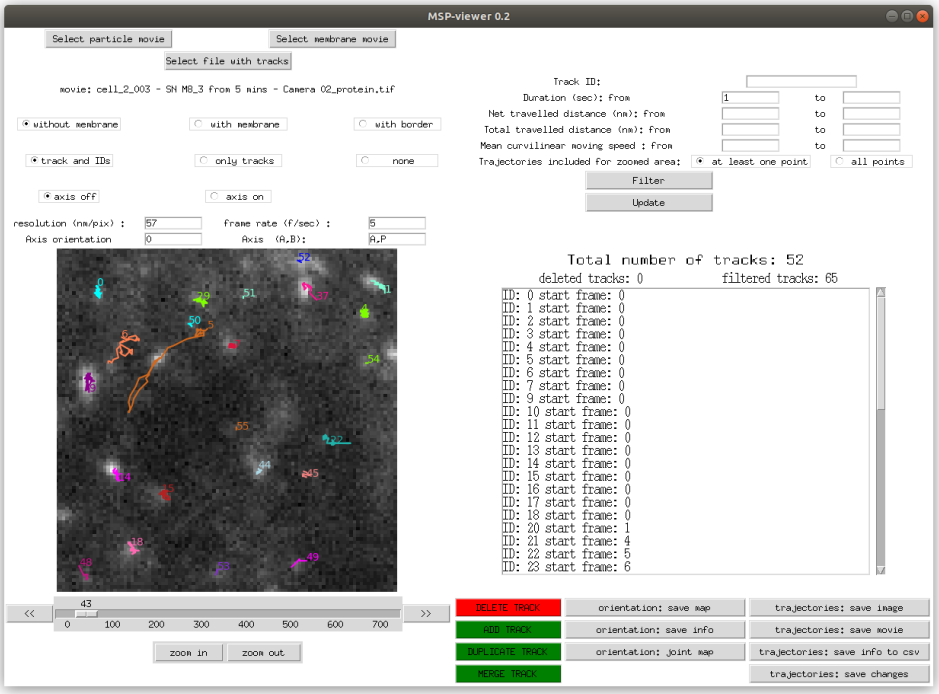
3.1 Interface overview: main window

The MSP-viewer is a tool for exploring already tracked trajectories. It allows the user to view the tracks, correct them if required, plot them and extract a number of characteristics.

3.1.1 Load the data

The image sequence with particles of interest should be a single channel 8bit tiff stack, trajectories can be loaded from a csv-file or txt-file with json format. Both formats are provided by the MSP-tracker. The membrane sequence can be loaded if required. Both image sequences, with particle and with segmented membrane, should have the same length and size (width and height). Use "select image sequence", "select file with tracks" and "select membrane sequence" to load the data. At last, provide the image sequence resolution and frame rate, otherwise, default parameters will be applied.

Figure 5: MSP-viewer interface



3.1.2 View trajectories

The Image viewer displays image sequence with plotted trajectories. You can select to view trajectories with their ID, just trajectories itself or the original image: "track and ID", "only tracks" and "none" respectively. If the segmented membrane is loaded, it can be plotted on the top of the image sequence ("with membrane" option) or a border/skeleton of the membrane can be extracted and plotted on the top of the image sequence ("with border" option).

3.1.3 Trajectory filter

All the loaded set of trajectories can be filtered based on their position, length/duration, net and total travelled distances and mean curvilinear moving speed. The tracks can be filtered using zoom option, for this you can zoom on the image and press "filter" button. The software will filter out all the track which are not located in the visible zone. The filter can be set to include or exclude tracks partly appearing in the zoomed area (select from the options for the trajectory included for zoom area). The duration (in seconds), travelled distance (in nm) and mean curvilinear moving speed (nm/sec) is calculated based on the provided resolution and frame rate. You also can choose particular trajectory by providing its track ID. Press "Filter" button to apply the filter.

You can monitor the number of filtered tracks above the trajectory list. Empty the filter and press "Filter" button again to return to the original state.

3.1.4 Removing and adding tracks

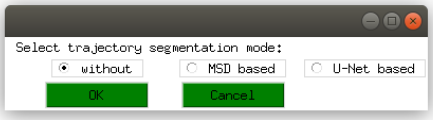
The existing tracks can be deleted, duplicated and merged. To delete a track you should first select it, press the button "DELETE TRACK" and confirm the action. The deleted track cannot be restored, but the changes will not be saved in the original file until you press "trajectories: save updates". You can monitor the number of deleted tracks above the trajectory list. To duplicate a track, select it and press "DUPLICATE TRACK" button, provide new track ID or use the suggest track ID value. To merge tracks use "MERGE TRACK" button and provide IDs of the tracks you want to merge. The merged tracks will not be deleted, the new merge track will appear with a new track ID.

A new track can be created with button "ADD TRACK". It can be filled with point (coordinates values and frames) in the individual trajectory window.

3.1.5 Export trajectory details for the entire image sequence

There are two options to visualise the tracking results: 1) plot all the trajectories in a single frame ("trajectories: save image") or 2) create an image sequence (movie) with plotted trajectories ("trajectories: save movie"). Deleted and filtered out trajectories will not appear in the exported image/sequence.

A number of parameters can be exported all together to a csv file ("trajectories: save info to csv"). The list of parameters can be found in Table 4. The particle speed is evaluated for the entire trajectory and for its moving segment. Therefore, you will be asked to select type of the trajectory segmentation:

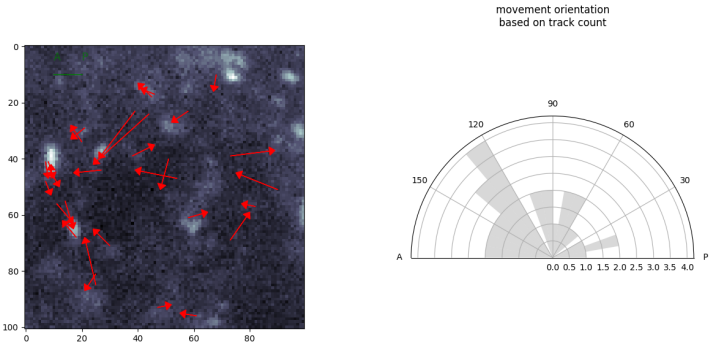


The choice "without" will not provide any information about the motion, "MSD based" will use Mean Square Displacement (MSD) to segment directed motion of the trajectory path, while "U-Net based" will provide segmentation using trained network. After selection of the trajectory segmentation mode, you will be redirected to select file name.

To save the changes use "trajectory: save update" button and provide a new filename or select existing one to overwrite it.

The MSP-tracker provides an option to plot a map with trajectory orientations (taking into account the first and the last coordinates of each trajectory) and a polar diagram of the orientations for the entire trajectory list, Figure 6. Use "orientation: save map" to create a png file with the orientation map and diagram of the current image sequences with loaded trajectories. If you want to plot results of a multiple sequences on the same polar diagram, firstly, save the orientation data with "orientation: save info" for each sequence, secondly, load all the saved orientation files together with "orientation: joint map". The software will ask the location for the plotted polar diagram.

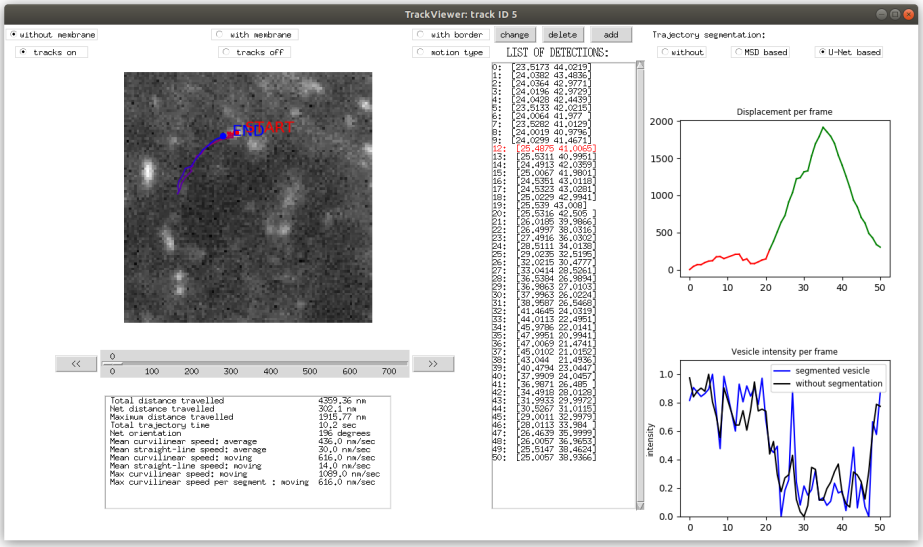
Figure 6: Trajectory orientation map and polar diagram



3.2 Individual trajectory window

The individual trajectory window can open by double click on the track ID in the trajectory list. The window provides the image viewer, a list of quantitative trajectory measures, a tool to correct the trajectory, a list of coordinates, and plots representing both trajectory displacement and changes in intensity of the particle along the trajectory. Red colour of the position in the coordinate list highlight discrepancies in frame order.

Figure 7: MSP-viewer: individual trajectory window



The image viewer provides an option to view the membrane (if one uploaded) and to visualise the trajectory itself. There are three preference: 1) "tracks on" option to view the trajectory itself (red colour represents start of the trajectory and blue the end point) 2) "tracks off" option to view the original frame, and 3) "motion type" to colour the trajectory based on the motion type (green - moving particle and red - stalled particle). The approach for the segmentation of the

trajectory path into different motion components can be selected on the right top side of the window. The choice "without" will not provide any information about the motion, "MSD based" will use Mean Square Displacement (MSD) to segment directed motion of the trajectory path, while "U-Net based" will provide segmentation using trained network.

Quantitative trajectory measures

There is a number of the quantitative measures evaluated for each trajectory. The descriptions of the measures are provided in Table 4. The speed parameters provided for entire trajectory (average) and for the segments where particle has a directed motion (moving).

Measure	Formulation
Total distance travelled	$d_{total} = \sum_{i=1}^{N-1} d(p_i + p_{i+1})$
Net distance travelled	$d_{net} = d(p_i + p_N)$
Maximum distance travelled	$d_{max} = max_i d(p_1 + p_i)$
Total trajectory time	$t_{traj} = (N-1)t$
Net orientation	α
Mean curvilinear speed	$\nu = \frac{1}{N-1} \sum_{i=1}^{N-1} \frac{d(p_i + p_{i+1})}{t}$
Mean straight-line speed	$\nu_{line} = \frac{d_{net}}{t_{traj}}$

Table 4: Quantitative trajectory measures

Correcting trajectory points

Trajectory can be corrected by adding, removing or changing the points/detections. To delete a point select it in the list of detections and press "delete" button. You can select multiple position in the list to delete them simultaneously.

Confirm the action in the appeared window. To add a new trajectory, just use "add" button and provide frame number and coordinates. You can click on the viewer to grab position and frame from the image sequence. Use "apply" to confirm the action and "apply and add" to confirm the action and start a adding one more detection.

To change coordinates value in a particular point - select the point in the list and press "change". If multiple positions are selected only the earliest one (with the smallest frame number) will be changed. The coordinates can be specified by typing them manually or click on the appeared image window.

Displacement and intensity plots

Two plots are provided to represent trajectory characteristics: displacement plot (on the top) and intensity plot (on the bottom).

The displacement plot represents a displacement of the particle in relation to the first point of the trajectory. The color of the line highlights the motion type, where green represents moving particle segment and red - stalled(not moving) particle.

The intensity is plotted for two cases - segmented particle and intensity of the ROI (region of interest). Both intensities are normalised within the trajectory points. It allows the user to analyse changes in particle intensity.

4 References

Nguyen 2013: Uyen T.V. Nguyen, Alauddin Bhuiyan, Laurence A.F. Park, Kotagiri Ramamohanarao. An effective retinal blood vessel segmentation method using multi-scale line detection. Pattern Recognition, Volume 46, Issue 3, 2013, Pages 703-715.
Dmitrieva 2019: M. Dmitrieva, H. L. Zenner, J. Richens, D. S. Johnston and J. Rittscher, "Protein Tracking By CNN-Based Candidate Pruning And Two-Step Linking With Bayesian Network," 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), Pittsburgh, PA, USA, 2019, pp. 1-6, doi: 10.1109/MLSP.2019.8918873.