**Explanation of program main\_tracking.m**

The program is modular and consists of the main file (main\_tracking.m) which calls separate functions. All these functions are quickly described in the following, and the parameters which influence the performance of these functions are given plus a note where to change them.

After running of main\_tracking.m, all parameters, setup, and results are saved in a Matlab Data file 'Results' in the created Results folder. These files can be opened and inspected in Matlab.

If you have any questions/comments/suggestions, please email Laura Weimann (laura.weimann@gmail.com).

**1) The first function to be called is:**

**[setup]=spot\_detection\_real\_data(parameters,stack\_directory);**

**Description**: It reads in TIFF video files, bandpass filters the files, and identifies spots in the filtered image data of each frame. The image of the first image frame (filtered data) pops up, and identified spot candidates are shown as red crosses. Then the corresponding raw data image pops up, with all spots which passed the SNR and size threshold. SNR values of the filtered and raw data are plotted onto the image. Spots are then localised in each frame via calculating their centroid position

**Results**: in the ‘Results SPT Analysis’ folder a subfolder with the experiment name is created, which contains a subfolder called ‘spot detection Results’, in which the coordinates of all identified spots are saved in .dat files. Per spot, 6 data entries are saved: x,y coordinate, Max Intensity filtered data, Max Intensity raw data, SNR filtered data, SNR raw data. A copy of the analysed image stack is saved in this folder as well.

Further, 6 control plots for the respective image stack are created and saved in the same folder: A histogram of the maximum intensity for all identified spots (1) raw and 2) filtered data), a histogram of the SNR for all identified spots (3) raw and 4) filtered data), the number of spots identified in the respective image frame over the frame number 5) after applying the initial intensity threshold and 6) after SNR and size exclusion.

Further, in the ‘spot detection Results’ folder 3 results plot comparing the respective image stacks are saved: 1) The absolute threshold (mean(image\_stack) + Parameters.initialthreshold\*std(image\_stack), with image\_stack containing the intensity values of the respective stack) is plotted over the image stack, 2) the mean value of the spots identified in the first image frame (raw data) is plotted over the image stack, 3) the mean particle-particle distance of particles identified in the first image frame is plotted as a function of the image stack.

**Output**: setup is a Structure Array containing the following information

|  |  |
| --- | --- |
| M | Image Height |
| N | Image Width |
| K | # of images in video |
| density | Mean value of nearest neighbour distances of spots identified in the first image frame, first value refers to first video etc |
| directory | Directory of folder in which spot detection results are being saved |
| SNR | SNR of all identified spots in the first frame, SNR = (max\_intensity-mean(background))/(sqrt(std(background)^2)); The background is defined as the 20\*20 pixel image corners in the first filtered image frame. Applied to filtered data |
| SNR\_raw | Same as SNR, but applied to raw data |
| SNR\_mean | Mean value of SNR\_raw |
| spots\_initial\_th | # spots identified in all frames after initial thresholding |
| spots\_second\_th | # spots identified in all frames after exclusion based on SNR and size |
| threshold | Absolut threshold value used for spot identification, see Parameters.initialthreshold below |
| stack\_directory | Directory of folder in which data to be analysed is |

**Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Where to change | Description | Recommendation |
| Parameters.lobject | Header of main\_tracking.m | defines parameter for bpass function to cancel out the long wavelength noise | Integer length in pixels somewhat  larger than a typical object. |
| Parameters.lnoise | Header of main\_tracking.m | defines parameter for bpass function to cancel out the short wavelength noise | Characteristic lengthscale of noise in pixels. Additive noise averaged over this length should vanish. Set to 1. |
| Parameters.initialthreshold | Header of main\_tracking.m | threshold for spot detection, number of std deviations above mean of background of filtered image. Background is defined as the mean intensity value of the whole image stack. | A low value will slow down analysis, a high value may miss true events. Try something between 3 and 10, and check the thresholding visually. |
| Parameters.pkfnd\_sz | Header of main\_tracking.m | defines diameter of spots in which only brighter one will be selected | Odd integer only, cntrd\_sz – 2 |
| Parameters.cntrd\_sz | Header of main\_tracking.m | defines diameter of area of spots for centroid calculation defines | Odd integer only, number should be big enough to capture the whole particle but not so big that it captures others. A larger value makes the localisation more accurate. (recommended: 5,7) or even larger if density is very low |
| parameters.max\_spot\_size | Header of main\_tracking.m | Maximum Spot Size [pixel]  Which is still accepted | 10-100 |
| parameters.SNR = 2 | Header of main\_tracking.m | Minimum SNR of the filtered data which is still accepted | 2-10, depends on data quality |

**2) The second function to be called is:**

**[results] = tracking\_laura\_real\_data(parameters,setup);**

**Description**: It reads in .dat files of the spot coordinates and connects particles in subsequent frames with their nearest neighbours. All particle trajectories regardless of the image frame in which they appear are recovered, and particle blinking can be taken into account as well (set parameters.memory accordingly, see below).

**Results**: In the respective results folder creative by function 1) another folder is created, called 'Results Tracking' which contains .dat files of all particle trajectories (6 entries: image frame, x,y coordinate, Max Intensity raw data, SNR filtered data, SNR raw data) . A matlab figure (Trajectory Ensemble) showing all trajectories is created and an .avi file (track.avi) showing the tracking results on raw data. The number refers to the trajectory number. Two control histograms are created and saved in the respective folder: 1) a histogram of the track lengths [frames] and 2) of the mean SNR raw data per trajectory.

**Output**: results is a Structure Array containing the following information

|  |  |
| --- | --- |
| meanSNRrawpertrack | Mean SNR per trajectory, raw data |
| meantracklength | Mean length of trajectories |

**Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Where to change | Description | Recommendation |
| parameters.minLength | Header of main\_tracking.m | defines minimal length of tracks to be kept [frames] | 3-10 |
| Parameters.max\_step | Header of main\_tracking.m | defines maximal distance at which 2 spots are linked in subsequent frames [pixels] | an estimate of the maximum distance that a particle would move in a single time interval |
| Parameters.memory | Header of main\_tracking.m | # of time steps a particle can be lost and recovered again | 0-4, depends on the data quality |
| Parameters.withvideo | Header of main\_tracking.m | Will create .avi video file if set to 1 |  |
| Parameters.fbs | Header of main\_tracking.m | defines Frames per second for output videos | 10 |

**3) [results] = msd\_main\_laura\_real\_data(parameters,param\_guess1,param\_guess2,param\_guess3,results,setup);**

**Description:** This function performs an MSD and subsequently a JD analysis on the trajectory data. The MSD analysis yields a single MSD value for each trajectory, whereas the JD analysis analyses an ensemble of trajectories. The advantage of the JD analysis is that it allows to resolve different mobility populations in the data. For more information see: <http://www.plosone.org/article/authors/info%3Adoi%2F10.1371%2Fjournal.pone.0064287;jsessionid=074F3CF35E284B5DDC8D6037A82D4542>

**Results:** In the ‘Results SPT Analysis/experiment name’ folder two subfolders, MSD Analysis Results and JD Analysis Results are created.

MSD Analysis Results:

MSD-values per trajectory are obtained and summarised in a histogram, on a normal and on a logarithmic scale. The MSD intercept can be used to estimate the localisation precision obtained, and a histogram of all standard deviations as a measure for the localisation precision is obtained. Further two scatterplots are created and saved, testing for a potential correlation between the respective standard deviation/Diffusion coefficient and the respective trajectory length. Finally, the average over the MSDs of all trajectories is obtained and an ensemble MSD value is obtained by fitting the average MSD using a weighted linear fit (MSD\_mean).

JD Analysis Results:

Histograms and cumulative histograms of jump distances are obtained and the fit results are saved. Depending on the setting of parameters.number\_populations, 1-3 mobility populations are fitted to the data and the resulting figures are saved.

**Output:** results is a Structure Array containing the following information

|  |  |
| --- | --- |
| meanSNRrawpertrack | Mean SNR per trajectory, raw data |
| meantracklength | Mean length of trajectories |
| D\_JD\_1pop | Diffusion coefficient assuming only 1 mobility population |
| f\_JD\_1pop | Fraction of particles in mobility mode 1, 1 by definition |
| D\_JD\_2pop | Diffusion coefficients assuming 2 mobility populations |
| f\_JD\_2pop | Fraction of particles in mobility modes defined by D\_JD\_2pop |
| D\_JD\_3pop | Diffusion coefficients assuming 3 mobility populations |
| f\_JD\_3pop | Fraction of particles in mobility modes defined by D\_JD\_3pop |
| jump\_distances | All jump distances obtained from the trajectory data |
| D\_MSD | Mean diffusion coefficient of all individually obtained diffusion coefficients, as obtained by MSD analysis |
| D\_ensemble | Diffusion coefficient obtained by averaging MSD-values over whole ensemble, and subsequent linear fit |
| diff\_coefficients | All diffusion coefficients obtained from the trajectory data by means of the MSD approach |
| n\_cell | # video stacks for analysis |
| n\_tracks | # trajectories for analysis |

**Parameters: MSD Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Where to change | Description | Recommendation |
| parameters.threshold\_tracklength | Header of main\_tracking.m | Defines minimum trajectory length for tracks considered for MSD analysis | 5-10 |
| Parameters.step | Header of main\_tracking.m | Number of time points for calculating MSD and for linear fitting. | A small value (2-5) yields short term diffusion coefficients, large values yield long-term diffusion coefficients, set to approximately of the average trajectory length |
| parameters.n\_fit | Header of main\_tracking.m | Number of points for fitting the ensemble plot averaged over all trajectories | Should be smaller or equal to parameters.step |
| parameters.bool\_D | Header of main\_tracking.m | If set to 1, for every single trajectory Parameters.step is determined individually by step = trajectory\_length/4. If trajectory\_length/4 < 3, step is set to 3.  If set to 0, the same step as defined by Parameters.step is used. |  |
| parameters.show\_single\_MSD\_plots |  | If set to 1, the linear fits of all trajectories are shown | Set to 1 only if total numbers of trajectories is small, otherwise too many Matlab figures are created. |

**Parameters: JD Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Where to change | Description | Recommendation |
| parameters.number\_populations | Header of main\_tracking.m | Number of mobility populations expected in the data | 1-3, fitting for values larger than 4 not implemented |
| param\_guess1(1) | Header of main\_tracking.m | Expected value for diffusion coefficient D of population [in mu^2/s] | 0.01 – 1 |
| param\_guess2(1) | Header of main\_tracking.m | Expected value for diffusion coefficient D of first mobility population [in mu^2/s] | 0.01 – 1 |
| param\_guess2(2) | Header of main\_tracking.m | Expected value for particle fraction of first mobility population | 0.1-0.9 |
| param\_guess2(3) | Header of main\_tracking.m | Expected value for diffusion coefficient D of second mobility population [in mu^2/s] | 0.01 – 1 |
| param\_guess3(1) | Header of main\_tracking.m | Expected value for diffusion coefficient D of first mobility population [in mu^2/s] | 0.01 – 1 |
| param\_guess3(2) | Header of main\_tracking.m | Expected value for particle fraction of first mobility population | 0.1-0.9 |
| param\_guess3(3) | Header of main\_tracking.m | Expected value for diffusion coefficient D of second mobility population [in mu^2/s] | 0.01 – 1 |
| param\_guess3(4) | Header of main\_tracking.m | Expected value for particle fraction of second mobility population | 0.1-0.9 |
| param\_guess3(5) | Header of main\_tracking.m | Expected value for diffusion coefficient D of third mobility population [in mu^2/s] | 0.01 – 1 |