Documentation of Matlab functions in the Single Particle Tracking and Analysis toolbox

The following documentation is meant for users dealing with single particle data who want to extract statistical information of single particle trajectories. All code is written in Matlab R2013a on a Windows 7 OS. The functions are pretty much modular and can be used as is or may need some augmentation for different single particle data sources.

Yonatan Golan, 2017
golanyoni@gmail.com

Functions documentation

1. Raw data extraction and single particle tracking
    1.1. ***ThunderStorm_From_Matlab*** – Initiates a ThunderSTORM instance from within Matlab. This is useful for batch data extraction. The function receives the path of a .nd2 file and creates a molecular list as a .csv file in the same directory.
    Explicit Dependencies: ImageJ, ThunderSTORM plugin, Bio-Formats plugin and a Windows OS for some system commands. This can probably be adjusted for other operations systems.

    1.2. ***CsvNd2_to_Tracks*** – Generates single particle trajectories from a molecular list. The function retrieves the time vector from the given .nd2 and a molecular list from the corresponding .csv file and creates a .mat file containing the resulting trajectories.
    Explicit Dependencies: *TimeStamp_From_ND2.m, ReadMolList_ForTracking.m, Track_FromMolList.m, Track_Lengths.m*

    1.3. ***TimeStamp_From_ND2*** – Reads an ND2 file and retrieves the time vector from it. This is relevant for determining the frame rate independently or having the whole time vector if it is non-consistent. The function uses the Bio Formats matlab directory which can be downloaded from http://www.openmicroscopy.org/site .
    Explicit Dependencies: bfGetReader.m (from the Bio Formats Matlab directory).

    1.4. ***ReadMolList_ForTracking*** – Reads a molecular list from a .csv file and arranges it in a way to fit the track.m tracking algorithm.
    Explicit dependencies: *ReadMolList.m*

    1.5. ***ReadMolList*** – Reads a molecular list from any text file and any headers. It can read the desired headers or open a dialogue in which one can choose the desired headers to read.

    1.6. ***Track_FromMolList*** – A wrapper function for the track.m function written by DanielB and EricD. See http://site.physics.georgetown.edu/matlab/.
    Explicity dependencies: *track.m*

    1.7. ***Track_Lengths*** – Returns the duration of all tracks and a list of tracks in a given duration range.

    1.8. ***Tracks_To_Cells*** – Converts the struct containing all particle data into cell arrays, each containing a single trajectory. This is useful for further analysis.

2. Segmentation of mobility states
    2.1. ***Segment_All_Trajectories*** - Segments all given trajectories according to the Cummulative Escape Radius (CER) method.
    Explicit dependencies: *Track_Mobility_by_CER.m, Split_Mobility_Multi.m*
    2.2. ***Segment_All_Trajectories_vbHMM*** - Segments all given trajectories according to the vbSPT method (Perrson et.al 2013). Very similar to *Segment_All_Trajectories,* except, it uses the mobility results obtained from the vbSPT method instead of CER.
    Explicit dependencies: *vbSPT package, Split_Mobility_Multi.m*

2.3. ***Track_Mobility_by_CER*** – Perform the CER classification method. Assign a mobility value for each time step of the trajectory according to the maximal distance traveled from it in a given time window.
Explicit dependencies: *Eliminate_singular_events.m*

2.4. ***Split_Mobility_Multi*** – Split trajectories according to their calculated mobility states.

2.5. ***Eliminate_singular_events*** – Smooth out singular events of opposite mobility. For example a mobility pattern of a trajectory: 0001000 will be smoothed out to 0000000. Longer singular events can also be smoothed out according to the given input.

2.6. ***Multi_Gaussian_Fitting*** – Fit multiple data sets (X's and Y's) with two or three Gaussian components. The function finds the parameters of the best fitting Gaussian (either 2 or 3) for the whole data set. This function is useful for determining the efficiency of a given segmentation process.
Explicit dependencies: *lsqcurvemultifit.m*

2.7. ***lsqcurvemultifit*** **-** solves non-linear least squares problems for multiple data sets. The function makes use of Matlab's lsqcurvefit function. It feeds it concatenated datasets for finding a common fit for all of them.