## Process Description:

"Tracking" takes in the objects identified in the detection step and links them together to construct their trajectories over time.

***Input Channels:***

This allows you to select the channels containing the detected objects you want to track. Select the channels by clicking on them in the "Available Input Channels" box and then clicking "Select>" to move them to the "Selected Channels" box. You can unselect a channel by clicking the "Delete" button.

***Parameters:***

**Problem Dimensionality**: Choose 2 or 3 from the drop-down menu.

**Maximum Gap to Close**: Longest gap resulting from temporary particle disappearance that the tracker will attempt to close.

**Minimum Length of Track Segments From First Step**: This parameter sets the minimum number of consecutive frames over which an object must be tracked for it to be used in the gap closing.

Check "**Do segment merging**" to look for merging events.
Check "**Do segment splitting**" to look for splitting events.

Check "**Plot histogram of gap lengths …**" to view the number of closed gaps of various lengths. This plot helps in assessing whether the "maximum gap to close" parameter is good or too large (in which case the histogram of gap lengths will plateau for longer gaps).

Check "**Show calculation progress in command line**" to see calculation progress in the matlab command line. This will show 3 iterations of the frame-to-frame linking step (forward, backward and forward) and then the gap closing, merging and splitting step. The frame-to-frame linking step is done in 3 iterations in order to optimize the self-adaptive tracking parameters learned while tracking.

***Cost Function:***

Choose from the drop down menus the functions to calculate the costs in the frame-to-frame linking step (Step 1) and the gap closing, merging and splitting step (Step 2).

Currently there are two options for each step:
Step 1: **Brownian + Directed motion models** or **Microtubule dynamics**
Step 2: **Brownian + Directed motion models** or **Microtubule dynamics**

These cost functions use per-particle motion modeling, with the following possible models:
(1) Brownian motion.
(2) Motion along a straight line with constant velocity.
(3) Motion along a straight line with constant velocity and possible instantaneous directional reversal.

Motion modeling is implemented via the Kalman filter, whose functions are defined in the next block.

Click on “**Setting**” to define the parameters for each cost function.

### *Kalman Filter Functions:*

Choose the set of Kalman filter functions for motion modeling. Currently there are two options: **Brownian + Directed motion models** or **Microtubule dynamics.**

Click on “**Setting**” to define the parameters for the initialization function, if desired.

### *Input and Output*

Define name and location of .mat file where results will be saved.

Choose “**Export tracking result to matrix format**” to convert the tracks from the default structure format to a matrix format – see format definitions below. In this case, there will be an additional results file, stored in the same location as the main results file. It will have the same name as the main results file, with “_mat” appended to its end.

NOTE: The matrix format can be large, so be cautious in choosing this option when analyzing large movies. Furthermore, merging and splitting information will be lost in the matrix format.

Choose “**Apply settings to all movies**” if you are analyzing multiple movies and you want to analyze them all using the same parameter settings.

OUTPUT FORMAT:

<u>Structure format:</u>

The default output of the tracker is a structure array, called tracksFinal, stored in the .mat file specified above. If the tracker finds **N tracks** in the movie, tracksFinal will be an **N×1 structure array** (every element corresponds to one track). These tracks are **compound tracks**. If, for example, two particles first move separately and then they merge, their tracks will appear together as one compound track entry in tracksFinal.

Every entry in tracksFinal (i.e. every compound track) contains **3 fields**:

**(1) tracksFeatIndxCG**: Connectivity matrix of particles between frames, after gap closing.
Number of rows = Number of tracks merging with each other and splitting from each other (i.e. involved in compound track).
Number of columns = Number of frames the compound track spans.
Zeros indicate frames where a track does not exist, either because those frames are before the track starts or after it ends, or because of temporary particle disappearance.

**(2) tracksCoordAmpCG**: The positions and amplitudes of the tracked particles, after gap closing.
Number of rows = Number of tracks merging with each other and splitting from each other (i.e. involved in compound track).
Number of columns = 8 × number of frames the compound track spans. For every frame, the matrix stores the particle's x-coordinate, y-coordinate, z-coordinate (0 if 2D), amplitude, x-coordinate standard deviation, y-coordinate standard deviation, z-coordinate standard deviation (0 if 2D) and amplitude standard deviation.
NaNs indicate frames where a track does not exist, either because those frames are before the track starts or after it ends, or because of temporary particle disappearance.

**(3) seqOfEvents**: Matrix storing the sequence of events in a compound track (i.e. track start, track end, track splitting and track merging).
Number of rows = number of events in a compound track.
Number of columns = 4.
In every row, the columns mean the following:
1st column indicates frame index where event happens.
2nd column indicates whether the event is the start of end of a track. 1 = start, 2 = end.
3rd column indicates the index of the track that starts or ends (The index is "local", within the compound track. It corresponds to the track's row number in tracksfeatIndxCG and tracksCoordAmpCG).
4th column indicates whether a start is a true initiation or a split, and whether an end is a true termination of a merge. In particular, if the 4th column is NaN, then a start is an initiation and an end is a termination. If the 4th column is a number, then the start is a split and the end is a merge, where the track of interest splits from / merges with the track indicated by the number in the 4th column.

Matrix format:

If "**Export tracking result to matrix format**" is checked, the tracks will be stored in matrix format as well. In this case, all tracks will be placed in one big matrix.

In the case of no merges and splits, there will be two new matrices:
**-trackedFeatureInfo**: This is the equivalent of tracksCoordAmpCG above, but for all the tracks together.
**-trackedFeatureIndx**: This is the equivalent of tracksFeatIndxCG above, but for all the tracks together.

In the case of merges and splits, there will be two additional output variables. These are needed because in this case a compound track can contain multiple segments (that merge and/or split) and thus the relationship between row number in the big matrices (trackedFeatureInfo and trackedFeatureIndx) and the entry number in the structure array (trackFinal) is not known unless stated explicitly.
**-trackStartRow**: An array indicating the row in the big matrices storing the information of the first segment of each compound track.
**-numSegments**: An array indicating the number of segments belonging to each compound track.