README for running DC-MSS, the Divide-and-Conquer Moment Scaling Spectrum transient mobility analysis algorithm described in "Multi-step track segmentation and motion classification for transient mobility analysis" by Vega et al., Biophysical Journal 2018. https://pubmed.ncbi.nlm.nih.gov/29539390/.

Brief overview of algorithm

DC-MSS works in three steps: (i) it utilizes a local movement descriptor throughout a track to divide it into initial segments of putatively different motion classes; (ii) it classifies these segments via Moment Scaling Spectrum (MSS) analysis of molecule displacements; and (iii) it uses the MSS analysis results to refine the track segmentation and classification. See publication for details.

Software installation

The software has been tested on **Matlab R2015b** and **R2016b** on **Windows 64-bit** and **Linux 64-bit**. It has not been tested on any MAC OS X.

The software requires the following **Matlab toolboxes**:

- Signal Processing Toolbox (tested using v. 7.0)

- Statistics and Machine Learning Toolbox (tested using v. 10.0)

The download package contains all necessary code (but not the Matlab toolboxes) to run DC-MSS once added to your MATLAB search path. Additionally, the package contains a track file named **test_track.mat** to test the analysis.

Running the DC-MSS Code

Once the package has been downloaded, unzipped, and added to your MATLAB search path, the function **basicTransientDiffusionAnalysisv1** can be used to run DC-MSS.

The inputs required to run the code and resulting output are described below:

Synopsis

```
[transDiffResults,errFlag] = basicTransientDiffusionAnalysisv1(tracks,probDim,plotRes,peakAlpha);
```

Enter the above in the command line, replacing input placeholders with actual values. With the exception of the **tracks** input, default input parameters can be used by replacing the input placeholders with empty brackets (i.e. [ ]).

Input:

tracks: Tracks to be analyzed (**test_track** can be loaded from DC-MSS package and used as an example).

IMPORTANT NOTE: If using u-track package for tracking (Jaqaman et al., Nat Methods 2008), the tracks as output by u-track are already correctly formatted and can be directly input for transient diffusion analysis. Otherwise, tracks should be formatted as follows: matrix with number of rows = number of tracks and number of columns = number of frames × 8. For each row (i.e. track), X coordinates should be indices 1:8:end, and Y coordinates should be indices 2:8:end, as shown below. See function help for more details.

X1	Y1							X2	Y2
1.2	4.6	0	0	0	0	0	0	2.8	5.3

probDim: Dimensionality of input tracks. Default: 2 (which is really the only option as 3D analysis has not been implemented yet).

plotRes: Enter value of 1 to plot tracks colored by mobility type, 0 otherwise. Default: 0.

Mobility color coding:

Brown: immobile

Blue: confined diffusion

Cyan: free diffusion

Magenta: directed diffusion (i.e. diffusion with drift)

Black: unclassified

peakAlpha: Confidence level for choosing peaks when initially segmenting track. Default: 95 (as described in paper).

Output:

transDiffAnalysisRes: A structure array with number of rows = number of tracks (or compound tracks in the case of merging and/or splitting). It contains the field ".segmentClass," with number of rows = number of individual tracks in the compound track (which merge with and/or split from each other). In case of no merging or splitting, each compound track is one individual track and segmentClass will contain one row only. segmentClass itself contains the following fields:.

momentScalingSpectrum: (Number of classification segments)-by-(20+probDim) matrix, where each row contains the information for one classification segment, and the columns store the following:

(1) Start frame of segment

(2) End frame of segment

(3) Classification of segment: (0 = immobile, 1=confined, 2=free, 3=directed, NaN = unclassified)

(4) MSS slope leading to classification

(5-11) Generalized diffusion coefficients for moment orders 0-6

(12-18) Scaling powers for moment orders 0-6

(19) Normal diffusion coefficient (from MSD analysis)

(20) Confinement radius or localization precision, if segment is classified as confined or immobile, respectively (NaN otherwise)

(21/22/23) Coordinates of segment center (1D, 2D or 3D, depending on probDim)

Please ignore the other fields "momentScalingSpectrum1D" and "asymmetry". These parts of the analysis are not fully implemented yet, and are not part of the Vega et al. Biophys. J. 2018 paper.

errorFlag: 0 if function runs normally, 1 otherwise.

Visualizing the Output

The output of DC-MSS can be visualized immediately after analysis via the **plotRes** input described above. However, if you would like to visualize the output at a later time point, you can use the function **plotTracksTransDiffAnalysis2D**. Input and output for this function are shown below.

Synopsis

plotTracksTransDiffAnalysis2D(tracks,transDiffAnalysisRes,timeRange,newFigure,image)

Enter the above in the command line, replacing input placeholders with actual values. With the exception of the **tracks** and **transDiffAnalysisRes** inputs, default input parameters can be used by replacing the input placeholders with empty brackets (i.e. [ ]).

Input:

tracks: Tracks previously analyzed with **basicTransientDiffusionAnalysisv1**, in same format as input to **basicTransientDiffusionAnalysisv1**.

transDiffAnalysisRes: Output of **basicTransientDiffuisonAnalysisv1** corresponding to the supplied **tracks**.

timeRange: 2-element row vector indicating time range to plot, [start,end]. Optional. Default: whole movie.

newFigure: 1 if plot should be made in a new figure window, 0 otherwise (in which case it will be plotted in an existing figure window). Optional. Default: 1.

image: An image that the tracks will be overlaid on if newFigure=1. It will be ignored if newFigure=0. Optional. Default: no image.

Output:

The plot

Mobility color coding

Brown: immobile

Blue: confined diffusion

Cyan: free diffusion

Magenta: directed diffusion (i.e. diffusion with drift)

Black: unclassified

Simulating Tracks

While a sample track has been provided, additional tracks can be simulated using the function **simMultiMotionTypeTraj** as described below. Note: only the output **trackMatrix** is needed to run DC-MSS. The output trajTransDiffClass can then be compared to the results from running DC-MSS.

Synopsis

[traj,trackMatrix,trajTransDiffClass,errFlag]=simMultiMotionTypeTraj(numParticles,volumeEdges,

totalTime,timeStep,diffCoefRange,confRadRange,driftVelRange,durationRange,strictSwitch,

immPrecision)

Enter the above in the command line, replacing input placeholders with actual values, as described below.

Input:

numParticles: Number of particles in simulation.

volumeEdges: Edges of volume in which particles reside. A row vector with number of entries = system dimensionality (e.g. [512, 512] for a 512x512 area).

totalTime: Total time of simulation [time units].

timeStep: Simulation time step [time units].

diffCoefRange: Row vector with 2 entries indicating diffusion coefficient range [(space units)^2/(unit time)].

confRadRange: Row vector with 2 entries indicating confinement radius range [space units].

driftVelRange: Row vector with 2 entries indicating drift speed range [space units/unit time]. Direction chosen randomly by algorithm.

durationRange: 4-by-2 array indicating range of time spent in each motion category [time units]. 1st row: confined diffusion; 2nd row: free diffusion; 3rd row: directed diffusion; 4th row immobility. 1st column: shortest duration per category; 2nd column: longest duration per category. To exclude a category, put zeros in its row.

strictSwitch: 1 to force switching to a different motion type between segments, 0 to allow switching to the same motion type but possibly with different parameters.

immPrecision: Localization precision for simulating immobile tracks. Default: 0.8 [space units].

In the above descriptions, time is in "time units" and space is in "space units." These could be whatever is relevant for the simulation, as long as the same time unit and space unit are used for all input variables. Examples: Time unit = frame or second, space unit = pixel or micrometer. The output tracks will be in the same space units as the input, and the time interval between time points will be the input timeStep.

Output:

traj: (Number of time points) - by - (dimensionality) - by - (number of particles) array of particle trajectories.

trackMatrix: Trajectories in matrix format, as described for the input of **basicTransientDiffusionAnalysisv1**

trajTransDiffClass: Structure indicating trajectory segment diffusion classification. Format as described for the output of **basicTransientDiffusionAnalysisv1**.

errFlag: 0 if function executes normally, 1 otherwise.