

README for running diffModesF2F, a package for diffusion mode analysis based on frame-to-frame displacements. The algorithms are described in “Changes in single-molecule integrin dynamics linked to local cellular behavior” by Jaqaman et al., Molecular Biology of the Cell 2016 and in “Heterogeneity in VEGF Receptor-2 mobility and organization on the endothelial cell surface leads to diverse activation models by VEGF” by da Rocha-Azevedo et al. 2020.

Brief overview

diffModeF2F calculates the diffusion coefficient of a track from its frame-to-frame displacements, taking into account the track’s localization precision. Supplying a “diffusion mode divider” structure that defines specific diffusion modes (each mode defined by a range of diffusion coefficients) allows also the classification of each track into one of the specified diffusion modes based on its calculated diffusion coefficient.

Software installation

The software has been tested on **Matlab R2018b** on **Linux 64-bit OS**.

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

- Statistics and Machine Learning Toolbox
- Control System Toolbox
- Optimization Toolbox
- Signal Processing Toolbox
- Image Processing Toolbox
- Computer Vision System Toolbox
- Curve Fitting Toolbox
- Parallel Computing Toolbox
- Matlab Distributed Computing Server

Please also down the u-track particle tracking package (<https://github.com/DanuserLab/u-track>), as some functionalities in the diffModesF2F package require functions in the u-track package. The two packages together contain all necessary code (but not the Matlab toolboxes) to run diffModesF2F once added to your MATLAB path.

Running the code

The documentation describes two scenarios. Scenario 1 describes how to run the core diffusion mode analysis function. Scenario 2 describes how to run the diffusion mode analysis code within an object-oriented movie data management infrastructure. For this, the movies will have to be analyzed (i.e. particle detection and tracking) via u-track, in order to set up the object-oriented framework as part of the tracking process.

Scenario 1: Calling the core diffusion mode analysis function

```
diffModeAnalysisRes = trackDiffModeAnalysis(tracksFinal,diffModeDividerStruct);
```

The first input argument are the tracks to be analyzed. They must be in the format of the tracks as output by u-track. Please see u-track documentation for details.

The second input argument is a variable that allows track classification into diffusion modes, based on the supplied divider(s) and each track's diffusion coefficient. More on this below.

If the diffusion mode divider structure is not available, the analysis code can be called without it:

```
diffModeAnalysisRes = trackDiffModeAnalysis(tracksFinal);
```

In this case, each track's diffusion coefficient will be calculated, but the tracks will not be classified into diffusion modes.

The output of trackDiffModeAnalysis, diffModeAnalysisRes, is described in detail in the function "trackDiffModeAnalysis."

*** More on the diffusion mode divider(s):

Two diffusion mode divider examples are given in the "ExampleData" subdirectory of diffModesF2F. diffModeDividerStructExample1 contains the diffusion mode divider used in the VEGFR-2 study (da Rocha-Azevedo et al. 2020). For VEGFR-2 we found 2 diffusion modes, and thus it needed 1 divider between them. diffModeDividerStructExample2 contains the diffusion mode dividers used in the integrin study (Jaqaman et al. 2016). For integrins we found 4 modes, and thus they required 3 dividers. Also, the dividers used in the integrin study were more sophisticated, and depended on track length and localization precision. See publications for details.

IMPORTANT: Other molecules will have other diffusion properties and will need different diffusion mode dividers. Here we describe two possible ways to obtain the diffusion mode dividers (there could be others as well):

(1) In a first run, trackDiffModeAnalysis can be called without the diffusion mode dividers, in order to get the distribution of diffusion coefficients of the molecule of interest. From this, the diffusion modes and their divider(s) can be derived. The function can then be re-run with the dividers as input, in order to classify the tracks. An example of this approach is described in the VEGFR-2 study (da Rocha-Azevedo et al. 2020).

(2) The package contains another function, getDiffModes, which derives a molecule's diffusion modes from the distribution of its frame-to-frame displacements. Simulations can be then used to determine the dividers between the modes, with which one can then call the function trackDiffModeAnalysis. An example of this approach is described in the integrin study (Jaqaman et al. 2016). See also the help block of "getDiffModes" for details.

Scenario 2: Diffusion mode analysis within the object-oriented movie data management infrastructure

Step 1. Perform particle detection and tracking using uTrackPackageGUI. Please refer to u-track documentation for help.

Tracking with uTrackPackageGUI places each movie and its analyses within an object called “movieData.” The GUI also allows one to put multiple movieData objects together in a new object called “movieList.” This allows the diffusion mode analysis software to use the movieData and movieList infrastructure, which makes the analysis process both more efficient and more organized.

Step 2. Perform diffusion mode analysis.

Function calls:

- Open movieData or movieList in workspace:
For movieData: MD = MovieData.load;
For movieList: ML = MovieList.load;
- Load the variable “diffModeDividerStruct,” similar to the examples in diffModeDividerStructExample1.mat and diffModeDividerStructExample2.mat (located inside diffModesF2F/ExampleData directory). See Scenario 1 for more information about this variable.
- Call analysis code:

If diffModeDividerStruct exists:

```
analyzeDiffusionModesMLMD(ML/MD,[],[],0,[],[],[],[],diffModeDividerStruct,0);
```

If diffModeDividerStruct does not exist:

```
analyzeDiffusionModesMLMD(ML/MD,[],[],0,[],[],[],[],[],0);
```

The first input is the ML or MD variable opened in the workspace (not both).

The output of analyzeDiffusionModesMLMD is saved in a directory called DiffusionModeAnalysis belonging to each analyzed movie, as part of the movie data management infrastructure.

Calling analyzeDiffusionModesMLMD with the above input arguments is equivalent to calling the core function in Scenario 1. Please see the help of “analyzeDiffusionModesMLMD” for a description of its input and output, and the other analysis possibilities that it offers (including calling getDiffModes to determine the diffusion modes, if desired).