Manual for analytical diffusion distribution analysis (anaDDA)

Author(s): J.N.A.Vink

Version 2.0

Last update: 09.02.2021

Installation

Clone the anaDDA folder from Github to the desired folder on your desktop. Subsequently add this folder and its subfolders to the search path of MATLAB.

MATLAB version and required toolboxes

This software was tested on MATLAB 2018a and 2020a. The Statistics and Machine Learning Toolbox is required to run the software. In case the user wants to use parallel computing for increased performance, the Parallel Computing Toolbox is also required.

Test example file

To test whether anaDDA is functioning correctly on your desktop type 'run anaDDA' and press 'Enter' in MATLAB. Running this should bring up a prompt which allows the user to enter and modify parameters that are linked to the type of measurement and data that the user wants to fit. For now, the user can leave the default parameters as they are. The section 'Description Input Parameters' will provide more details on the parameters

After pushing 'Ok', a second prompt allows selecting a data file. This should be the dataset that you want to fit. There are two example data files included in the anaDDA folder: 'ExamplefileD.mat' and 'Examplefiletracks.mat'. These represent the two different input formats that anaDDA can handle. The section "Input data" will provide more details on the used data format. For now, select 'ExamplefileD.mat'.

After selecting the input file, the program will start to run. The command window will show the progress that the program is making in finding a fit. The MLE algorithm is designed to have a minimum number of runs (default 4) with different starting parameters to avoid running into local minima. Subsequently, the two best runs are compared and if the parameters are close enough to each other, the algorithm stops and uses the best run to continue the program. If the two best parameter sets are still different the (indicating that a global minimum might not have been found yet), the algorithm keeps running more cycles with different starting parameters until the best parameter set has been recovered twice. With the example file ('ExamplefileD _koff20_kon20_Dfree4.mat'), the parameters should converge quickly to k_{off} = 20.3 s^{-1}, k_{on} = 20.7 s^{-1}, D_{free} = 4.07 $\mu m^2/s$ which are close to the values used in the simulation to create this dataset (k_{off} = 20 s^{-1}, k_{on} = 20 s^{-1}, D_{free} = 4 $\mu m^2/s$). The input parameters for the example tracks file ('Examplefiletracks _koff100_kon80_Dfree6.mat') were k_{off} = 100 s^{-1}, k_{on} = 80 s^{-1} and D_{free} = 6 $\mu m^2/s$.

The output of this run is an '_outputanaDDA_*datetime*.txt' file, in which the found parameters and estimated bootstrap standard deviations for each species are stored. A second file is generated which is the inputfile, which contains the parameters that were used during the run, it's named

'ExampleFileD_outputanaDDA _*datetime*_inputfile.mat'. This file allows the user to rerun the anaDDA with the same input parameters on different datasets by running anaDDA with the inputfile (e.g. anaDDA(inputfile)). Furthermore, plots are printed on screen that show the fit of the theoretical anaDDA distribution to the dataset that was provided for all the different number of track lengths that were in this dataset (up to 8 steps).

Input data

anaDDA is able to use either a list of *D* values or a list of tracked localizations. The format of these two inputs is described below.

1) list of *D* values

The input from a list of *D* values is a matrix consisting of 3 columns and *n* rows where n is the number of tracks. The first column is the calculated mean diffusion coefficient of a track (data from two-dimensional tracking). The mean diffusion coefficient for each track is calculated from the diffusion coefficients of all jump distances in between two frames within that track. The second column is the number of steps of the track (ranging from 1-8) and the third column is the frame time used in the measurement (in s). A step is a distance between two subsequent localizations, so for tracks with step number 4, the number of localizations within those tracks is 5. If individual tracks are longer than 8 steps, the *D* value should be calculated from the first 8 steps.

2) list of tracked localizations

The input from a list of tracked localizations is a matrix consisting of 5 columns and *n* rows where *n* is the number of localizations. The first column is the *x* coordinate (μm), the second column is the *y* coordinate (μm), the third column is the frame number, the fourth column is the track id (localizations belonging to the same track receive the same track id) and the fifth column is the frame time at which this track was measured (s).

Description Input Parameters

AnaDDA works with an input file for which (1) some parameters can be changed in the prompt if anaDDA is run without input, or (2) an input file can be generated beforehand which gives full control over all the parameters used in the program. Below you can find a description for each parameter in the input file that is generated by the script 'Generateinputfile.m':

Basic input parameters

numberofspecies (Default = 1)

This parameter controls how many species you want to include in the fitting. With species, we are referring to a collection of one or multiple states with different diffusion coefficients that can transition into one another. A species with more than two states can be approximated as two species depending on whether the transitions are slow enough compared to the frame time (see Manuscript).

upperDfree (Default = 10 μm^2/s)

Maximum of estimated D_{free} (μm^2/s). Do not set too high as a much higher value than the estimated *D* impacts precision/performance.

sigmaerror (Default = 0.03 μm)

The estimated localization error (μm) of the measured data. This localization error will also be added to simulated localizations if simulations are run. If you want to fit the localization error, choose fitlocerror = 1 (see below) or set localization error = -1 in the prompt window.

confinement (Default = false)

If true, the algorithm will assume that the tracks are confined by spherical or rod-shaped cell boundaries and change the predicted distribution based on the input parameters *lengthcell* and *radiusofcell*.

radiusofcell (Default = 0.5 μm)

The radius of the cells/confinement boundaries (μm). Only required if *confinement* is set to true. *l* This cell radius will also be used in simulations.

lengthcell (Default = 3 μm)

The length of the cell that you want to use in simulation (μm). Only required if *confinement* is set to true. This length is defined for spherical and rod-shaped cells. In case of spherical cells, the length cell should be twice the size of the radius of the cell. In case of rod-shaped cells, the length of the cell is the distance between the two poles.

compensatetracking (Default = false)

If true, the algorithm will assume that the tracks were analyzed with an implemented tracking window and change the predicted distribution based on the input parameter *trackingwindow*.

trackingwindow (Default = 1 μm)

The length of the tracking window used in the analysis (μm). Only required if *compensatetracking* is set to true.

fixedparameters (Default = [-1 -1 -1 -1 0; -1 -1 -1 -1 0; -1 -1 -1 -1 0] (no fixed parameters))

This parameter controls which input kinetic parameters are fixed. Each species has a separate row and each column defines a parameter. The respective order is fraction, k_{off} (s^{-1}), k_{on} (s^{-1}), D_{free} (μm^2/s) and D_1

(μm^2/s) for each column. Each of these parameters can be fixed independently. When you fit only one species, you only have to change values in the top row, for two species only the top two rows are used. For parameters that you do not want to fix, you supply the value -1. For D_1 the diffusion coefficient of the slowest state, the default value is set to 0, which assumes that the slowest state is immobile.

fitlocerror (Default = 0)

If one, the localization error will be fitted to the data. Does not work together with fitting of D1 (5th column of fixedparameters == -1).

Advanced input parameters fitting

bootstrapping **(**Default = true)

This parameter controls whether to include bootstrapping in your analysis. Bootstrapping will not change the output of the fit but will give an estimate of the standard deviation of the predicted output parameters.

numberofbootstraps (Default = 8)

Number of bootstraps that are used to create the estimated bootstrap values.

cyclenumber (Default = 4)

The minimum amount of runs of MLE you want to perform before comparing the best two and testing for convergence.

lowerstartkoff (Default = 0.05 s^{-1})

The lower bounds for the k_{off} parameters (s^{-1}) that are fitted during MLE optimization.

upperstartkoff (Default = 2000 s^{-1})

The upper bounds for the k_{off} parameters (s^{-1}) that are fitted during MLE optimization.

lowerstartkon (Default = 0.1 k_{off})

The lower bounds for the k_{on} parameters that are fitted during MLE optimization. In this case the k_{on} is calculated with respect to k_{off} so this parameter represents k_{on} = *lowerstartkon** k_{off}. This generates starting parameters that converge faster.

upperstartkon (Default = 10 k_{off})

The upper bounds for the k_{on} parameters that are fitted during MLE optimization. In this case the k_{on} is calculated with respect to k_{off} so this parameter represents k_{on} = upperstartkon* k_{off}. This generates starting parameters that converge faster.

precision (Default = 2^16 points)

Number of points for which the distribution is directly calculated. The likelihood of other points are derived from these points via interpolation. This parameter will determine the runtime and memory usage of your algorithm with higher precision leading to longer runtime and more memory usage. The algorithm is more efficient (due to FFT convolution) if precision is a multiple of 2.

nofit (Default = false)

If you do not want to fit the data/simulation but directly want to compare the data to the distribution made from the k_{off}, k_{on} and D_{free} parameters supplied in the input file select true.

plot (Default = true)

If you do not want to output figures, set to false.

KSStats **(**Default = true)

If you do not want to output KSStatistics of the fit, set to false.

Integrationinterval **(**Default = 200)

Number of points calculated via integration of PDA statistics with the exponential decay function. Rest of points is calculated with interpolation and convolution.

Advanced input parameters simulations

koff1 (Default = 30 s^{-1})

The off-rate of the first state of each simulated species (s^{-1}). In case *nofit* is set to true, this parameter is also used directly to generate the predicted anaDDA distribution.

koff2 (Default = 10e9 $s^{-1)}$

The off-rate of the second state of each simulated species (s^{-1}). If no second state exists, choose a much higher value than k_{off1}. In case *nofit* is set to true, this parameter is also used directly to generate the predicted anaDDA distribution.

kon1 (Default = 30 s^{-1})

The on-rate of the first state of each simulated species (s^{-1}). In case *nofit* is set to true, this parameter is also used directly to generate the predicted anaDDA distribution.

kon2 **(**Default = 0.0001 s^{-1})

The on-rate of the second state of each simulated species (s^{-1}). If no second state exists choose a much lower value than *kon1*. In case *nofit* is set to true, this parameter is also used directly to generate the predicted anaDDA distribution.

Dfree (Default = 4 $\mu m^2/s$)

The free diffusion coefficient of each simulated species. In case *nofit* is set to true, this parameter is also used directly to generate the predicted anaDDA distribution.

D1 (Default = 0 $\mu m^2/s$)

The diffusion coefficient of the slowest state. Default this is 0, which is the case for particles with an immobile state.

fraction (Default = 1)

Fraction of each simulated species. In case *nofit* is set to true, this parameter is also used directly to generate the predicted anaDDA distribution.

Nparticles (Default = 50.000 particles)

Number of particles (tracks) you want to simulate.

distributionNparticles **(**Default = [0.28723 0.20581 0.14747 0.10567 0.07572 0.05425 0.03887 0.08496] representing an exponential decay with mean length of 3)

Distribution of track lengths for the number of particles. In experimental data the distribution of track lengths usually follows an exponential decay with a lot of short tracks and a small number of large tracks. To capture this in simulation you can set the *distributionNparticles* which is the fraction of tracks for each track length (ranging from 1:8 steps). Make sure that the sum of these values is equal to 1.

frametime **(**Default = 0.01 sec)

The frametime (sec) you want to use in simulation.

framerange **(**Default = [1:8])

The range of track lengths (number of steps) that you want to use in simulation and subsequently fit with anaDDA.

steptime **(**Default = 1e-5 s)

The time steps that are made in the simulation. For fast transition rates and higher confinement levels the duration of the steptime needs to be sufficiently short to achieve accurate simulation.

frametimerange **(**Default = 0.01 s)

The different frame times for which tracks are simulated (s). In case that more than one value is provided (e.g. *frametimerange* = [0.01 0.05]), the same number of tracks are simulated for each frame time.

density (Default = 0)

Used to simulate tracking errors. This parameter determines how many particles on average are present at a given time point. For density combine with below parameter. If 0, than no tracking errors.

fovsize (Default = 40)

Used to indicate the size of the FOV in which the particles are placed. To calculate the density of particles use density/fovsize^2

Run simulations

The user can also run their own simulations with the included simulation scripts in the package and subsequently test anaDDA given certain kinetic parameters. The desired input parameters can be changed in the Generateinputfile script before it is run (e.g. input.koff1_A = 50 s^{-1} and input.kon1_A = 25 s^{-1}) for any of the parameters described above. Then generate this input file by running 'input = Generateinputfile'. For a single simulation the user can subsequently run 'Comparesimulationwiththeory(input)' which will start a simulation.

Alternatively, for a range of kinetic parameters the user can open Varyparameterssimulationcomparison and adjust the values at the top of the script. The range of parameters is subsequently simulated and tested with anaDDA by running 'Varyparameterssimulationcomparison(input)'.