Function **classification_on_file** is used to classify trajectories loaded from a .mat file

The function assumes the input comes in the form x,y,z,...,N where N is the trajectory serial number, starting from one.

Input:

file - string containing the file name, ending with .mat

Outputs:

prediction - Classification to diffusion model type where 0=FBM; 1=Brownian; 2=CTRW

y_full - matrix of network probabilities. Each trajectory recieves 3 values

which are the probabilities of being assigned to a specific

diffusion model.

Function **classification_heatmap** is used to test the classification network on FBM or CTRW trajectories as a function of noise levels and motion type parameters

The function plots the error fraction (1-accuracy) heatmap.

Input:

motionType - paramter to select motion model; 0=FBM, 1=CTRW

noiseType - parameter to select noise model; 0=Gaussian, 1=Ornstein-Unghlebek noise

Outputs:

accuracy - Fraction of successful classifications

prediction - vector of network predictions

Function **confusion_matrices** generates simulated training data with various SNR levels and prints out a confusion matrix per SNR value.

Input:

noiseType - parameter to select noise model; 0=Gaussian, 1=Ornstein-Unghlebek noise

Outputs:

None

Function **Brownian_predict_1D** is used to run a version of the network on a single trajectory.

The function assumes the input is a column vector

Input:

x - 1D column vector containing localization data

stepsActual - number of steps to analyze in each trajectory.

pixelSize - size of 1 pixel (in nm) of the experimental system

reg_model - loaded model of the network (comes as input to avoid reloading the same model for each trajectory)

Outputs:

pX - Network prediction for the Hurst exponenet of the 1D trajectory.

Function **Brownian_on_file** analyses data from a file and estimates diffusion coefficients per trajectory.

Input:

file - string containing the file name, ending with .mat

stepsActual - number of steps to analyze in each trajectory.

pixelSize - size of 1 pixel (in nm) of the experimental system

dT - time difference between subsequent steps

Outputs:

prediction - Network estimation of the diffusion coefficient

Dmsd - Temporal mean square displacement estimation of the diffusion coefficient

Function **Brownian_net_on_sim** analyses simulated trajectories and estimates diffusion coefficients per trajectory.

Input:

param - Diffusion coefficient of the simulated trajectories (units of [um^2/sec])

stepsActual - number of steps to analyze in each trajectory.

pixelSize - size of 1 pixel (in nm) of the experimental system

dT - time difference between subsequent steps

Outputs:

prediction - Network estimation of the diffusion coefficient

Dmsd - Temporal mean square displacement estimation of the diffusion coefficient

Function **MT_net_on_file** receives a .mat file containing trajectories of N-dimensions organized as x,y,z,..N where N is a trajectory serial number starting from 1.

NOTE: The net assumes all trajectories are of length 10, shorter trajectories will cause an error

The function constructs a matrix suitable for network input and analyzes the data.

Input:

file - string containing the file name, ending with .mat

Outputs:

prediction - A vector with length as the number of dimensions in the data,

containing network analysis results.

Function **ensemble_net_histogram_plot** generates 200 matrices of N trajectories with

Hurst exponents in the range [param-0.05,param+0.05] and analyses them using

the MultiTrack (MT) network.

The network receives data in the shape: [1,numTraj,10,1]

Input:

param - Hurst exponent ground truth

sig - 1/SNR

NTraj - number of trajectories in a single matrix (a number from 10,20,30,..150)

Outputs:

Hnet - Hurst exponent estimations by the network

Hmsd - Hurst exponent estimations by time averaged MSD

Function **ensemble_net_estimation_rmse** is designed to test the effects of more trajectories on estimation performance. The network iterates over the number of trajectories used M. For each value of M, 500 matrices of Mx10 are generated and analyzed.

The Function plots RMSE between estimation and ground truth as a function of number of trajectories and Estimation mean and std as a function of number of trajectories.

Input:

param - Hurst exponent ground truth

sig - 1/SNR

Outputs:

rmseMSD - RMSE values of MSD estimation to ground truth

rmseNET - RMSE values of network estimation to ground truth

aVec - MSD estimations of all 500 matrices for each number of trajectories

HNetVec - Network estimations of all 500 matrices for each number of trajectories

Function **predict_1D** is used to run a version of the network on a single trajectory.

The function assumes the input is a column vector

Input:

x - 1D column vector containing localization data

stepsActual - number of steps to analyze in each trajectory. This number determnines what network will run on the data.

Select from options: [10,25,60,100,200,300,500,700,1000]

reg_model - loaded model of the network (comes as input to avoid reloading the same model for each trajectory)

Outputs:

pX - Network prediction for the Hurst exponenet of the 1D trajectory.

Function **FBM_net_on_file** is used to run a version of the network on a .mat file containing one or more single particle trajectories.

The function assumes the input comes in the form x,y,z,...,N where N is the trajectory serial number, starting from one.

Input:

file - string containing the file name, ending with .mat

NTraj - number of trajectories to analyze; input '0' to analyze all trajectories

stepsActual - number of steps to analyze in each trajectory. This number determnines what network will run on the data.

Select from options: [25,60,100,200,300,500,700,1000]

Outputs:

prediction - A vector with lenght as number of trajectories containing network predictions (average of N-dimensional predictions)

NDpred - A matrix with dimensions [#trajetories,#dimensions] containing
all predictions done by the network (N-dimensions for each trajectory)

Function **net_on_mat** is used to run a version of the network on a numpy array containing one or more single particle trajectories. The function assumes the input matrix has the form:

[number of dimensions][number of trajectories][number of steps]

Input:

X - data input matrix

NTraj - number of trajectories to analyze; input '0' to analyze all
trajectories

stepsActual - number of steps to analyze in each trajectory. This number
determnines what network will run on the data.
Select from options: [25,60,100,200,300,500,700,1000]

Outputs:

prediction - A vector with length as number of trajectories containing
network predictions (average of N-dimensional predictions)

NDpred - A matrix with dimensions [#trajetories,#dimensions] containing
all predictions done by the network (N-dimensions for each trajectory)

Function **ST_nets_heatmaps** generates FBM trajectories with random Hurst parameters, and tests network performance on varying levels of noise and for different lengths of trajectories

Input:

numTraj - number of FBM trajectories to generate, default is 100

Outputs:

RMSE_net - A matrix containing RMSE values between net prediction and ground truth
as a function of noise level and trajectory length

RMSE_msd - A matrix containing RMSE values between MSD prediction and ground truth
as a function of noise level and trajectory length

Function **temporal_MSD_compare** generates 1000 FBM trajectories with a given Hurst parameter, and a given noise level, and tests varying network performances against MSD.

Input:

parameter - Hurst exponent ground truth

sigma - 1/SNR where SNR is the desired noise level

Outputs:

Hmsd25/Hmsd100/Hmsd1000 - MSD estimations based on 25/100/1000 steps

Hnet25/Hnet100/Hnet1000 - Network estimations based on 25/100/1000 steps

Function **estimation_convergence** analyses data from a file and estimates confidence intervals as a function of number of trajectories analysed, for both network and ensemble MSD. Intervals are estimated using a form of bootstrapping.

Input:

file - string containing the file name, ending with .mat

stepsActual - number of steps to analyze in each trajectory. This number

determnines what network will run on the data.

Select from options: [25,60,100,200,300,500,700,1000]

Outputs:

EMSD_err - confidence intervals as a function of trajectories analyzed (EMSD analysis)

net_err - confidence intervals as a function of trajectories analyzed (net analysis)

EMSD_plot - Anomalous exponent estimation as a function of trajectories analyzed (EMSD analysis)

net_plot - Anomalous exponent estimation as a function of trajectories analyzed (net analysis)