Manual

A manual for maximum likelihood analysis of 2- and 3-color fluorescence photon trajectories of transition path (TP) data collected from immobilized or freely diffusing molecules using a confocal microscope (Section 4) and customization guides for arbitrary models (Section 5).

Contents

1. Requirements

MATLAB (≥ 2019b) The analysis program has been developed and tested on 2019a and 2023a.

In order to use pre-compiled MEX (https://www.mathworks.com/help/matlab/matlab_external/introducing-mex-files.html) analysis functions, you need MS windows (x64) and Visual C++2022 Community (https://visualstudio.microsoft.com/downloads/). Otherwise, see Section 5 to compile the MEX functions on your system.

2. Installation

Simply unzip to extract files and add the file location to your MATLAB path (typically, it takes less than a minute). For the examples in the script files, copy functions in 'private' directory to one of your MATLAB path.

3. Input files

The photon trajectories need to be formatted in a specific way.

3.1. Preprocessing of photon trajectories: identification of single transitions

The raw data is records of individual photon's arrival time and color and additional delay time from a laser pulse when pulsed laser excitation is used. For the maximum likelihood analysis, photon trajectories need to be segmented into single transitions (e.g., folding, unfolding, binding) using an appropriate method such as the Viterbi algorithm.

3.2. Photon trajectories

A photon trajectory data of each transition consists of a 3 - 5-column matrix. For a 5-column matrix, the first column elements are transition data indices. The second column element is spared for bin indices of photons, which is not used in the analysis. The third column elements are photon arrival times in 100 ns unit. The fourth column corresponds to delay time from the laser pulse trigger signal, which is used for identification of photons by donor and acceptor 1 laser excitation in pulsed interleaved excitation (PIE). This is not used in the analysis of the data collected by continuous-wave (CW) mode laser excitation. The fifth column elements correspond to photon color or detection channel. A 3- or 4-column matrix has the same format of the last 3 or 4 columns of the 5-column matrix. Photon trajectories of all transitions are contained into a MATLAB cell array.

4. Parameter Optimization

4.1. 2-color CW excitation

Analysis of simulated photon trajectories of 2-color FRET protein folding transitions are contained in **Analysis_ProteinFoldingTP2color.m**.

1. Section 1 lists simulation parameters.
2. Section 2 optimizes parameters (**mlhTPABIsimfit0C.m**) and estimates errors (**mlhTPABIsimfit0errorC.m**) using an instantaneous transition model (i.e., transition path time, t_{TP} = 0).
3. Section 3 computes likelihood values with various TP times and the FRET efficiency (*E*) values of the TP (**mlhTPABIsimcalC.m**) using the parameters optimized with the instantaneous transition model in Section 2.
4. Section 4 optimizes parameters including the TP time and *E* of the TP (**mlhTPABIsimfitEintNintmEC.m**) and estimates errors (**mlhTPABIsimfitEintNintmEerrorC.m**) for a single-TP model with options of various number of intermediate states.
5. Section 5 optimizes parameters (**mlhTPABIsimfitEintParC.m**) and estimates errors (**mlhTPABIsimfitEintParerrorC.m**) for a multiple TP model.

The above MATLAB functions use **mlhrateeigtrpathrevcalvitABIC.m** and **mlhrateeigtrpathrevcalvitABIparC.m** with C++ libraries **mlhTPABIcal_MT** and **mlhTPgenABIcal_MT**, respectively, to calculate likelihood values.

4.2. 3-color CW donor excitation: coupled binding and folding

Analysis of simulated photon trajectories of 3-color FRET binding and folding transitions of an intrinsically disordered protein (IDP) are contained in **Analysis_IDPbindingTP2path3colorCW.m**. This data simulates an experiment of binding of a donor (D)- and acceptor 1 (A1)-labeled IDP to its acceptor 2 (A2)-labeled binding partner.

1. Section 1 lists simulation parameters.
2. Section 2 optimizes parameters (**mlhTP3cABIsimfit0C.m**) and estimates errors (**mlhTP3cABIsimfit0errorC.m**) using an instantaneous transition model (i.e., transition path time, t_{TP} = 0).

3. Section 3 computes likelihood values with various TP times and the 2-color FRET efficiency (*E*) values of the TP (**mlhTP3cABIsimcalC.m**) of the 2-color transition data with D and A1 (DA1, A2 not present) or with D and A2 (DA2, A1 not present) using the parameters optimized with the instantaneous transition model in #2.
4. Section 4 computes likelihood values with various TP times and the 3-color acceptor 1 fraction and acceptor 2 fraction values (*eps*1 and *eps*2) of the TP (**mlhTP3cABIsimintEcalC.m**) of the 3-color transition data with active D, A1, and A2 using the parameters optimized with the instantaneous transition model in #2.
5. Section 5 computes likelihood values with various TP times and the 3-color acceptor 1 fraction and acceptor 2 fraction values (*eps*1 and *eps*2) of the TP (**mlhTP3cABIsimintEcalindiC.m**) of individual 3-color transitions using the parameters optimized with the instantaneous transition model in #2.
6. Section 6 optimizes parameters (**mlhTP3cABIsimfitEintParC.m**) and estimates errors (**mlhTP3cABIsimfitEintParerrorC.m**) for a multiple TP model.

The above MATLAB functions use **mlhrateeigtrpathrevcalvit3cABIparC.m** with a C++ library **mlhTPgen3cABIcal_MT** to calculate likelihood values of 3-color transitions and use **mlhrateeigtrpathrevcalvitABIC.m** and **mlhrateeigtrpathrevcalvitABIparC.m** with C++ libraries **mlhTPABIcal_MT** and **mlhTPgenABIcal_MT**, respectively, to calculate likelihood values of 2-color transitions.

7. Section 7 determines the TP time and acceptor fractions (3-color data) or FRET efficiencies (2-color data) of a single TP (**mlhTP3cABIfit1TPindi.m** and **mlhrateeigtrpathrevcalvit3cintEABI.m**) and estimates errors (**mlhTPABIfit1TPindierror.m**).
8. Section 8 plots acceptor fractions and TP times of individual trajectories determined in Section 7. Copy and run the definition of initparams and LUbounds in Section 7 before running Section 8.

4.3. 3-color PIE

Analysis of photon trajectories of 3-color FRET transitions collected using pulsed interleaved excitation (PIE). A C++ library **mlhTPgen3cAlexABIcal_MT** was developed. Other optimization routines and analysis examples will be updated.

5. Custom models

To implement arbitrary model, you need to build your own function. For this purpose, you need additional requirements:

GSL library	For parameter optimization and linear algebra.
MATLAB library	To call C/C++ functions in MATLAB.
Visual studio 2022	Or any C/C++ compiler compatible with MATLAB.

5.1. Requirements

5.1.1. GSL library

The current version of the analysis functions was built with GSL version 2.4. You can download GSL from https://www.gnu.org/software/gsl/. For Windows OS, you can use a pre-built version for Visual Studio (https://www.bruot.org/hp/libraries/) or you can build GSL using Visual Studio https://github.com/BrianGladman/gsl.git.

5.1.2. MATLAB library

MATLAB library for MEX can be found at YOUR MATLAB PATHWAY /extern/lib. General C++ MEX Application guide can be found at https://www.mathworks.com/help/matlab/cpp-mex-file-applications.html

5.1.3 Visual Studio 2022

The current version of the analysis functions was built on Visual Studio 2022 (https://visualstudio.microsoft.com/downloads/).

5.2 Customization

Unzip the project file. You can find the source codes in "src" directory and MATLAB wrapper functions in "mWrapper" directory.

5.2.1. Visual Studio 2022

Modify the library paths in "x64MexPropertySheet.props". There are two paths to be modified:
C:\Program Files\MATLAB\R2019b This is your MATLAB2019 path.
C:\gsl\lib This is your GSL library path.

Replace these paths to your MATLAB and GSL installation paths.

5.2.2 Step-by-step guide for the installation and the path setting

1. Install MATLAB 2019 (for other MATLAB versions, please check MEX application documents for the corresponding version).
2. Install Visual studio 2022 community version from https://visualstudio.microsoft.com/downloads/
3. Install GSL library for Visual Studio 2022 from https://www.bruot.org/hp/libraries/
5. Modify paths in "x64MexPropertySheet.props" as described above.

5.2.3 How to introduce a new model

To introduce your own model, you need to check the following functions at least.

analysisThread This is the function for likelihood evaluation. Here you need to define your rate matrix and the way you evaluate the overall likelihood (including photon colors and number of states).

input from matlab This function assign MATLAB input double arrays into C/C++ variables. Also, there are some pre-defined constants for certain models.
init mlh vars This function initializes variables. Depending on a model vectors and matrices dimension need to be adjusted.

5.2.4 MATLAB version dependencies in Visual Studio

For MATLAB 2019 (and maybe newer versions), MW_NEEDS_VERSION_H needs to be defined in the headers for Visual Studio. And MEXFUNCTION_LINKAGE needs to be used to define the gateway function. In older versions, you can use _declspec(dllexport) to define the gateway function.