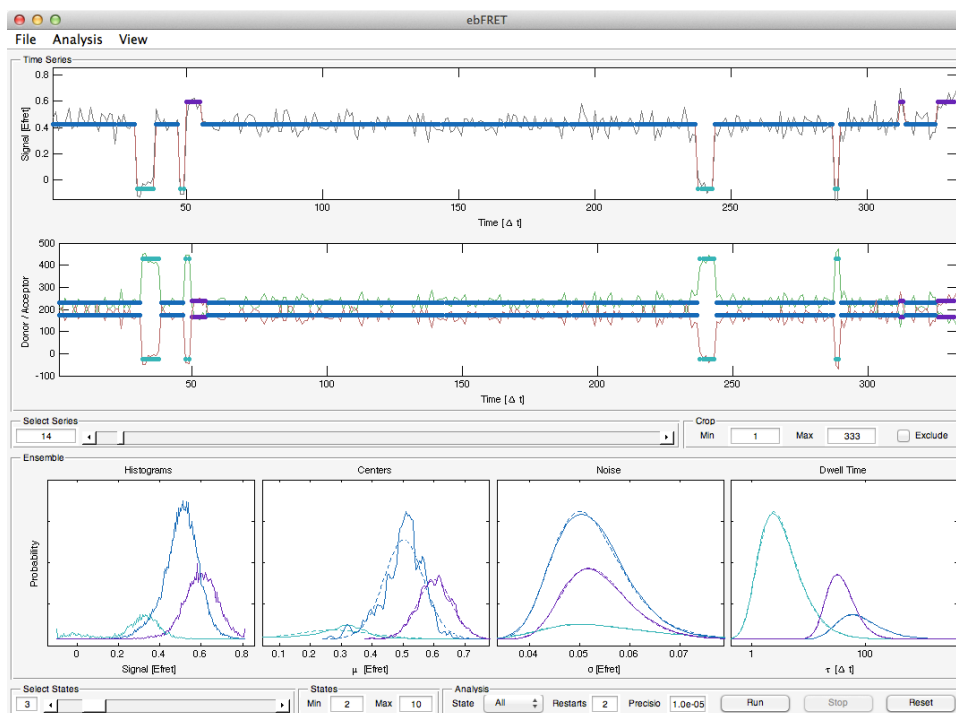


# ebFRET User Guide

Release 1.1, May 2014



## Contents

Getting Started . . . . .	2
Quick Usage . . . . .	2
System Requirements . . . . .	2
Installation . . . . .	3
Loading Data . . . . .	4
File Formats . . . . .	4
Loading Data from Multiple Files . . . . .	5
Grouping Files . . . . .	5
Performing Analysis . . . . .	6
Exploring the Data . . . . .	6
Pre-processing the Data . . . . .	6
Initializing the Priors . . . . .	7
Running Analysis . . . . .	9
Saving Results . . . . .	10
Advanced Topics . . . . .	12
Scripting the GUI . . . . .	12
Analysis with multiple CPUs . . . . .	14
Compilation of MEX files . . . . .	14

# Getting Started

## Quick Usage

1. Download the latest version of ebFRET from  
`https://github.com/ebfret/ebfret-gui/archive/master.zip`
2. Unzip the folder to some location, e.g.  
`C:\path\to\ebFRET`
3. Start up Matlab and add the ebFRET to the path by typing  
`addpath(genpath(C:\path\to\ebFRET))`
4. Open the GUI by typing  
`ebFRET()`
5. Load an example dataset by clicking File → Load and select the file  
`C:\path\to\ebFRET\datasets\simulated-K04-N350-raw-unstacked.dat`
6. Click Run to start analysis

## System Requirements

### Operating System

The ebFRET software suite should in principle run on all operating systems supported by Matlab (i.e. Windows, OS X and Linux). Development and testing was done with Matlab 2013a on OS X (Mountain Lion) and Windows 7.

### Matlab Version

We recommend using a Matlab version  $\geq 2012a$ . The GUI has been verified to work with Matlab versions as early as 2010b, but may exhibit performance issues related to the slower implementations of object-oriented programming methods in versions prior to 2012a.

### Memory Requirements

Memory requirements vary depending on the size of the dataset, the number of states, and the number of CPUs used. Typically, memory usage of a single instance should be less than 1 GB for moderately large datasets containing order  $10^5$  total datapoints in all time series.

### MEX Files

In order to achieve higher performance, some parts of the ebFRET code are written as MEX files in C. These functions may need to be compiled after installation. The ebFRET GUI attempts to do so automatically as needed, but this process will fail if the prerequisite compilers are not installed. In this event ebFRET will revert to using (slower) functions written in Matlab and print a warning message. For more information, see the *Advanced Topics* section.

## Installation

### Obtaining the Source Code

The latest version of the source for the ebFRET package may be obtained by visiting

<https://ebfret.github.io>

A zip file containing the sources can be found at

<https://github.com/ebfret/ebfret-gui/archive/master.zip>

Alternatively the source may be obtained using git

```
git clone git@github.com:ebfret/ebfret-gui.git
```

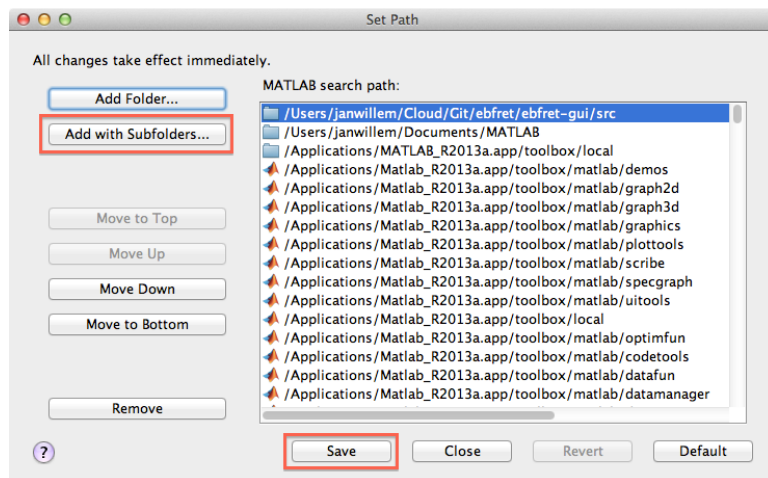
### Adding the ebFRET Source to the Matlab path

Once the ebFRET source file has been downloaded (and unzipped), it must be added to the Matlab path. This can be via

Environment → "Set Path" → "Add with Sub-folders" (Matlab ≥ 2013a)

File → "Set Path" → "Add with Sub-folders" (Matlab < 2013a)

Once the folder is added to the path, Save may be used to ensure the new path is retained in future Matlab sessions.



Alternatively, the folder containing the ebFRET source (e.g. C:\path\to\ebFRET) may be added by to the path via the command

```
addpath(genpath(C:\path\to\ebFRET))
```

### Starting the GUI

To open a new instance of the ebFRET GUI, simply type

```
ebf = ebFRET()
```

The returned object handle `ebf` is a matlab object that provides access to the entire GUI state. In particular, the loaded time series may be accessed via `ebf.series`, and the analysis results can be found in `ebf.analysis`. Accessing these values directly is not necessary in normal usage, but can be helpful if custom pre-processing or post-processing is needed. See the *Advanced Topics* section for further information.

## Loading Data

Data can be loaded into the gui via the `File` → `Load` menu item. ebFRET currently supports loading data from a number of different formats.

### File Formats

#### 1. ebFRET saved session (.mat)

Loading from this format restores an ebFRET session saved via `File` → `Save`. This file contains all data necessary to continue analysis from the point at which the file was saved.

#### 2. Raw donor-acceptor time series (.dat)

This loads a number of FRET time series from an ASCII file, which may be stored in two ways, which we call ‘stacked’ and ‘unstacked’. Example data stored in both formats can be found in the `examples` sub-directory of the ebFRET folder.

##### a. Stacked

The stacked format is useful for storing time series with variable length. Let’s assume we have  $N$  time series, with length  $T(n)$ , and intensities  $D(n, t)$  and  $A(n, t)$  for the donor and acceptor respectively. Each row in a stacked table now contains 3 values

label(1)	$D(1, 1)$	$A(1, 1)$
label(1)	$D(1, 2)$	$A(1, 2)$
...		
label(1)	$D(1, T(1))$	$A(1, T(1))$
...		
...		
label(N)	$D(N, T(N))$	$A(N, T(N))$

Where `label(n)` is a numeric unique id for each time series.

##### b. Unstacked

The unstacked format stores  $N$  times series of identical length  $T$  in  $T \times 2D$  matrix formatted as

label(1)	label(1)	...	label(N)	label(N)
$D(1, 1)$	$A(1, 1)$	...	$D(N, 1)$	$A(N, 1)$
$D(1, 2)$	$A(1, 2)$	...	$D(N, 2)$	$A(N, 2)$

$$\begin{array}{ccccc} \dots & \dots & \dots & \dots & \dots \\ D(1,T) & A(1,T) & \dots & D(N,T) & A(N,T) \end{array}$$

### 3. SFTracer donor-acceptor time series (.tsv)

This format is used by the SFTracer pre-processing suite for FRET time series, that is currently in preparation for release at the Gonzalez lab at Columbia. This data is stored as tab-separated values (TSV). The first two lines of each file contain meta-information and column headers respectively. The remaining lines store multi-color FRET data as

$$n \ c \ area(n) \ T(n) \ I0(n,c) \ I(n,c,1) \ I(n,c,2) \ \dots \ I(n,c,T(n))$$

Here  $c$  is the index of the color channel. The GUI assumes  $c=0$  for the donor signal and  $c=1$  for the acceptor signal.  $T(n)$  is the number of time points in time series  $n$ , and  $I0(n,c)$  is the background intensity of channel  $c$  for this time series.  $area(n)$  holds the surface area of the spot in the source image, which is currently ignored by ebFRET.

### 4. Single-molecule Dataset (.json, .json.gz, .mat)

The single-molecule dataset (SMD) format may be used to import data exported by other smFRET analysis packages, such as SMART<sup>1</sup>. A SMD file can be read from JSON (.json), gzipped JSON (.json.gz) and Matlab (.mat) formats, and can hold both raw (unanalyzed) data and unanalyzed results. Currently only raw time series can be read from SMD files. When loading SMD, the user may specify whether the stored time series contain Donor-Acceptor or FRET data. For each choice, the user must then select the SMD column names that contain the donor, acceptor, or FRET signals respectively.

## Loading Data from Multiple Files

When loading data from Raw or SFTracer formats, it is possible to select multiple files to load several datasets at once. Data can also be added in increments. The GUI will then prompt to either to either retain or replace the existing data.

## Grouping Files

The loading mechanism can implicitly be used to create ‘groups’ of files. For example, if one wishes to obtain separate parameter estimates for 3 sets of experiments, one can load each group separately by clicking File → Load. When data is loaded in this manner, the File → Export → “Analysis Summary” function (see *Saving Results*) will automatically calculate summary statistics separately for each group. If more sophisticated grouping criteria are needed, these must be implemented by hand. See the *Advanced Topics* section for further information.

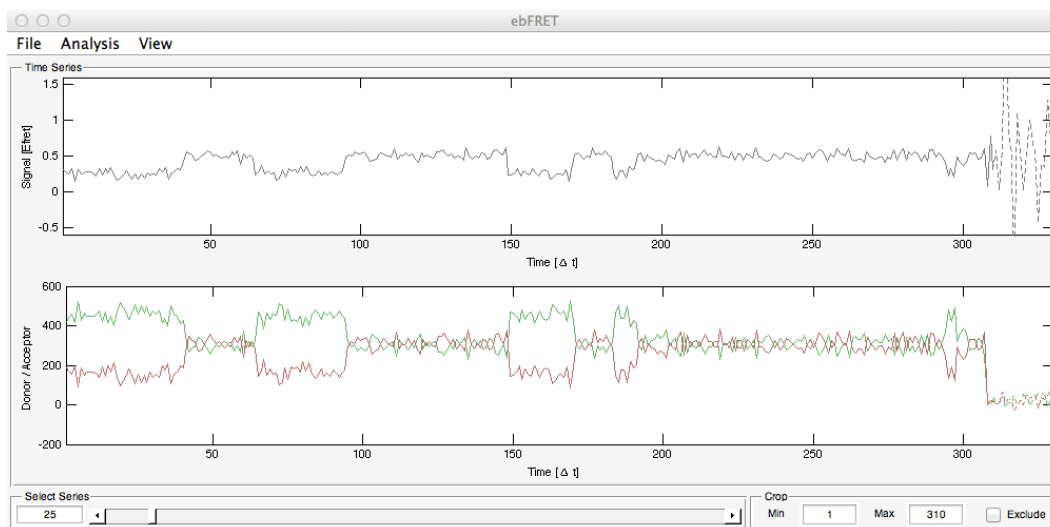
---

<sup>1</sup><https://simtk.org/home/smart>

# Performing Analysis

## Exploring the Data

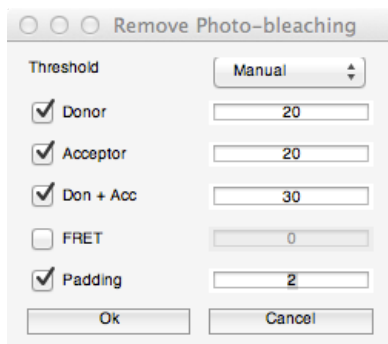
Once data has been loaded, the main window will display time series plots of the  $E_{\text{FRET}}$  ratio, as well as the donor and acceptor intensities (shown in green and red respectively).



The slider in the Series box underneath the plots may be used to view different time series in a dataset, and the edit box beside it may be used to select a specific series.

## Pre-processing the Data

### Photobleaching Detection



A common pre-processing task when working with FRET time series is cropping the time series to the point where either the donor or acceptor fluorophore photobleaches. ebFRET can do so according to several criteria using the dialog accessed via Analysis → "Remove Photobleaching"

#### 1. Manual

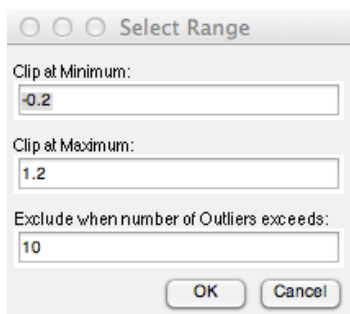
In this mode the time series is truncated when a selected value falls below the specified level. It is possible to select multiple criteria, in which case the time series is truncated

at the first point that satisfies one of the selected criteria. It is also possible to use a Padding to crop extra time points from the end of the series.

## 2. Automatic

In this mode, photobleaching is detected in a fully automated manner, by looking for jumps in the signal starting from the end of the time series.

### Clipping of Outlier Points



The image shows a software dialog box titled "Select Range". It contains three input fields: "Clip at Minimum:" with the value "-0.2", "Clip at Maximum:" with the value "1.2", and "Exclude when number of Outliers exceeds:" with the value "10". At the bottom of the dialog are two buttons labeled "OK" and "Cancel".

Time series in a dataset may exhibit  $E_{\text{FRET}}$  values that are either  $< 0$  or  $> 1$ . Note that such values can only occur when the acceptor or donor intensities falls below 0, which is by definition an unphysical occurrence that typically occurs when the background intensity cannot be determined accurately.

A small number of such outlier points can often significantly skew analysis outcomes. Because of this, it is generally a good idea to either clip outlier points  $\ll 0$  and  $\gg 0$  to more sanitized values, or exclude time series that exhibit many outlier points from the analysis entirely. This can be done by adjusting the minimum and maximum thresholds, as well as the maximum number of outliers.

### Manual Correction

Both photobleaching removal and detection of outliers are non-destructive operations. What this means is that all the data is retained in its original form, and simply pre-processed as needed when analysis is performed. Both operations can safely be run multiple times to find the best settings, without losing more and more data.

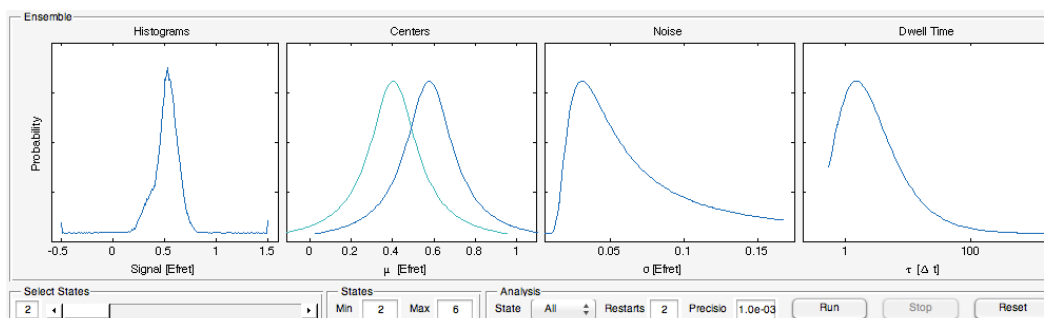
In addition there are Crop controls that can be used to correct the time interval that is used for analysis, as well as an Exclude button that can be used to entirely remove time series from analysis, or include a previously excluded time series once more.

## Initializing the Priors

### Automatic Initialization

In ebFRET multiple time series are analyzed to determine the 'range' of parameter values that best describes an ensemble of parameters. This range of values is described by a probability distribution known as the prior, and the goal when running analysis in ebFRET is to learn this prior from the data. In order to do so, the algorithm must be provided with an initial guess.

When the data is loaded, ebFRET makes an automatic guess for the initial values of the priors. In most case this guess will suffice. However, if there are outliers in the data, the algorithm for generating this initial guess may be a poor fit for the data. For this reason, the ebFRET GUI prompts the user to re-initialize the priors after photobleaching and outlier removal. In most cases the automatically determined result after post-processing should suffice.



After initialization, the Ensemble panel shows the distribution of measurement values and parameters. The first panel shows a simple histogram of measurement values, the second the distribution for state means, which should show a spread of values similar to the measurement histogram. The distributions on state noise levels and dwell times, shown in the 3rd and 4th panels respectively, are by default initialized with identical values.

## Manual Adjustment

If more precise control is needed, the priors may be adjusted manually using the dialog located under Analysis → "Set Priors"

The Min Center and Max Center parameters control the mean  $E_{\text{FRET}}$  values assumed for each state, which are spaced evenly between the extrema. The Noise parameter controls the expected emission noise around each state, whereas the Dwell parameter control how many time points are typically spent in one state, before a transition to a different state occurs.

In addition to the expected parameter values, it is also possible to specify a Prior Strength in terms of a number of equivalent observations. If this number is lowered, the prior distributions become more broadly peaked, and describe a larger range of allowable parameter



values. When this number is increased, the distribution becomes more tightly peaked, and a smaller range of parameter values is considered.

From experience we find that the initial guess for the prior generally should not strongly affect analysis outcomes. However, the Center Strength parameter can affect performance of the algorithm to some extent. If the Strength parameter is increased, the algorithm will be more likely to populate larger numbers of states. In this case the lower bound evidence must be used to decide among analysis results with different numbers of states. Setting the strength parameter to lower values will make the algorithm more likely to leave superfluous states unpopulated. This should result in consistent analysis outcomes as more states are considered, with correspondingly similar lower bound evidence values. We generally recommend setting the Center Strength parameter as low as possible, while making sure that no obvious features in the data are missed by the analysis.

Finally, priors may also be set directly from command window by manipulating values stored in `ebf.analysis.prior`. See the *Advanced Topics* section for further details.

## Running Analysis

Once data has been loaded, pre-processed, and the priors have been initialized, analysis can be started by pressing the Run button.

The ebFRET algorithm performs analysis by repeatedly executing two steps

1. Use variational Bayes (VB) estimation (i.e. the same method as vbFRET) to learn HMM parameters for each time series.
2. Use empirical Bayes (EB) estimation to adjust the prior distributions to match the posterior distributions learned for individual time series.

The settings that control how analysis is performed by the ebFRET GUI are

1. *States* → *Min, Max*

These parameters control the smallest and largest number of states considered.

2. *Analysis* → *State* → *All/Current*

When analysis is performed, it is possible to either run analysis only for the Current number of states, shown in the Ensemble view, or for all numbers of state, starting from the minimum.

3. *Analysis* → *Restarts*

This parameter controls the number of re-initializations used each time variational Bayes estimation is performed on an individual time series. If set to 0, VB estimation is performed only once, starting from the final parameters of the previous iteration. If set to 1, an additional ‘uninformative’ restart is performed. If set to values >1, further restarts are performed with random initializations.

Using more restarts makes the EB algorithm more likely to converge towards the correct result, but this comes at the expense of a computational cost proportional to the number of restarts. We recommend using 0-10 restarts, where 2 should be sufficient for most datasets.

#### 4. Analysis → Precision

This parameter controls the convergence threshold for both VB and EB estimation. Lowering this number yields more accurate results, at the expense of requiring a larger number of iterations. We typically recommend setting this value somewhere in the range  $1e-3$  to  $1e-6$ , where larger values should be used for quick interactive analysis, and  $1e-6$  represents a sensible setting for analysis that is to be included in a publication. In cases where a comparison is made between lower bound evidence values for different numbers of states, a rule of thumb is to run analysis long enough to obtain a difference between largest and next-to-largest values that is at least an order of magnitude larger than the convergence threshold.

## Saving Results

### ebFRET Session (.mat)

A ebFRET session can be saved to a Matlab file at any time via

File → Save

This can be used to store results and continue analysis at a later time. The Matlab file is also the only format that stores allowable data, including measurement values, settings for clipping, cropping, excluded traces, prior parameters, posterior parameters and viterbi paths. See the *Advanced Topics* section for a description of how data and analysis are stored.

### Export Analysis Summary (.csv)

A summary of analysis results can be exported with

File → Export → "Analysis Summary"

This creates a table in .csv format, which can be read in many other types of software, including Excel. Summary statistics are calculated by averaging over all time series, as well as any groups of time series that are defined. By default, time series in files that were loaded at once using File → Load are grouped together. In addition summary results are shown for analysis performed with different numbers of states, as well as each state within an analysis run.

The summary quantities calculated in this file are:

- Series
  - Basic statistics on the size of the datasets.
    - Number
      - Lists the total number of time series in each group.
    - Length
      - Lists the mean number of data points for time series in each group, along with the standard deviation, median, min, max and sum.
- Lower Bound
  - Information on the lower bound log evidence. A higher lower bound value indicates a better fit between model and data.

- Per Series  
Lists the mean lower bound for time series in each group, along with the standard deviation, median, min, max and sum.
- Per Observation  
Lists the mean lower bound per data point for time series in each group, along with the standard deviation, median, min, max and sum.
- Statistics  
Cumulative sufficient statistics over all time series. Averages calculated from the cumulative statistics weigh each time point equally, which is conceptually more or less equivalent to analyzing the results as if each time series represents a separate measurement interval of the same identical molecule.
  - Occupancy  
Lists the fraction and total number of time points assigned to each state.
  - Observation  
Lists the mean value and standard deviation of data points assigned to each state.
  - Transitions  
Lists the average transition rates and total number of transitions in all time series.
- Parameters  
These quantities list average parameter values predicted from the prior distribution that is learned from the data.  
The conceptual difference relative to the averaged statistics is that the prior is learned by weighing each time series, not each data point, equally. The mean parameter values therefore represent a mean over a population of molecules with varying physical properties, and their standard deviation represents the variability within this population.
  - Center  
Lists the mean and standard deviation of the state centers (i.e. the mean  $E_{\text{FRET}}$  value) predicted by the EB prior.
  - Precision  
Lists the mean and standard deviation and mode of the state precision (i.e.  $1 / \text{variance}$  of the  $E_{\text{FRET}}$  values) predicted by the EB prior.
  - Dwell Time  
Lists the mode of the state dwell time (i.e. the number of time points a molecule is expected to remain in a state, before transitioning to a different state) predicted by the EB prior.
  - Transition Rates  
Lists the mean and standard deviation of the transition rates predicted by the EB prior.

### Export Traces (.dat, .mat)

Time series may be exported via File → Export → "Traces", and can be stored in either an ASCII (.dat) or Matlab (.mat) format. The data stored is the index  $n$  of the time series, along

with up to five types of time series may be selected: the donor intensity  $d(n, t)$ , acceptor intensity  $a(n, t)$ , FRET ration  $f(n, t)$ , the state of the viterbi path  $z(n, t)$  and the corresponding FRET mean of the viterbi path  $\mu(n, t)$ .

The Viterbi path is the most likely sequence of conformational states, which after analysis can be calculated for each time series based on the learned prior and posterior parameters. Export of Viterbi paths is useful for comparing results with those from suites such as HAMMY<sup>2</sup> and vbFRET<sup>3</sup>, which perform independent analysis on each time series. Note that ebFRET, unlike HAMMY and vbFRET, does not need viterbi paths to calculate averaged kinetic rates over multiple time series. See the *Export Analysis Summary (.csv)* section for details.

This data is written out as a stacked set of time series. If all signal types are selected, this data contains the columns

1	$a(1, 1)$	$d(1, 1)$	$f(1, 1)$	$z(1, 1)$	$\mu(1, 1)$
1	$a(1, 2)$	$d(1, 2)$	$f(1, 2)$	$z(1, 2)$	$\mu(1, 2)$
...					
1	$a(1, T(N))$	$d(1, T(N))$	$f(1, T(N))$	$z(1, T(N))$	$\mu(1, T(N))$
...					
N	$a(N, T(N))$	$d(N, T(N))$	$f(N, T(N))$	$z(N, T(N))$	$\mu(N, T(N))$

### Export SMD (.json, .json.gz, .mat)

Time series and analysis results can be exported to a single-molecule dataset (SMD) format for interoperability with other software such as SMART. Currently, ebFRET can only export a single set of analysis results (using a fixed number of states, not a range), which must be selected by the user during export. Data can be written to JSON (.json), gzipped JSON (.json.gz) and Matlab (.mat) formats.

## Advanced Topics

### Scripting the GUI

All operations that can be performed in the ebFRET GUI can in principle also be scripted. In addition all data and parameters can be accessed and manipulated directly.

#### Time Series

Let's assume the GUI was started by calling `ebf = ebFRET()`. After loading a dataset, all time series are stored in a struct array `ebf.series`, allowing the  $n$ -th time series to be accessed via `ebf.series(n)`, which has the following fields

- donor

The uncropped donor signal

<sup>2</sup><http://bio.physics.illinois.edu/HaMMY.html>

<sup>3</sup><http://vbfret.sourceforge.net>

- `acceptor`  
The uncropped acceptor signal
- `fret`  
The uncropped, unclipped,  $E_{\text{FRET}}$  ratio.
- `crop`  
Defines the range of datapoints `min:max` to be included in analysis in terms of the bounds `ebf.series(n).min` and `ebf.series(n).max`.
- `exclude`  
Time series is excluded from analysis if set to `true`.
- `group`  
Integer index of group that time series is assigned to. When analysis results are exported using `File → Export → "Analysis Summary"`, each separate estimates are calculated for each group. By default, the ebFRET GUI assigns a new group index each time one or more files are loaded with `File → Load`.
- `file`  
Filename that time series was read from.
- `label`  
Label of time series, if it was specified in the file data was loaded from.

## Analysis Results

After performing analysis, all results of k-state analysis are stored in a struct array `ebf.analysis(k)`. This struct has the fields

- `dim`  
Specifies number of states
- `prior`  
Prior parameters learned from EB analysis.
- `posterior`  
Posterior parameters learned from EB analysis (for each time series)
- `expect`  
Expected values for sufficient statistics (for each time series)
- `viterbi`  
Viterbi sequence of most likely states (for each time series)
- `lowerbound`  
Lower bound on the log evidence (for each time series)
- `restart`  
Index of the selected randomized restart.

## Analysis with multiple CPUs

Analysis in ebFRET can be accelerated by using multiple CPUs. In order to do so, simply type

```
matlabpool('open', numworkers)
```

Here `numworkers` is the number of parallel Matlab instances one wishes to use, which typically should be set to the number of cores on the machine that is used. If the desired number of workers is more than the default maximum, this number can be changed via

Parallel → "Manage Cluster Profiles" → Edit → "Number of Workers ..."

*Note:* At the time of writing, there exists a known issue with the Java version included in OS X that causes the `matlabpool` command to fail<sup>4</sup>. A fix for this issue is available from Matlab<sup>5</sup>. There is also a workaround for this issue, allowing operation as normal by following this specific sequence of steps

1. Close and open Matlab
2. Open Parallel → "Manage Cluster Profiles"
3. Close the dialog and type `matlabpool('open', numworkers)`

## Compilation of MEX files

In order to obtain faster computational performance two parts of the ebFRET algorithm, are implemented in C as so-called MEX files. These are the `forwback` and `viterbi` procedures.

MEX files must be compiled before they can be used in Matlab. The ebFRET sources include pre-compiled binary files for Windows 7, Linux, and OS X. These binaries should in principle work out of the box on OS X and Linux, but Windows users may need to install the Microsoft Visual C++ 2010 Redistributable Package (x64), which can be downloaded at

<http://www.microsoft.com/en-us/download/confirmation.aspx?id=14632>

If binaries prove incompatible with the system, ebFRET will attempt to automatically recompile the MEX files. This requires a working compiler. For instructions see

<http://www.mathworks.com/support/compilers/R2013b/index.html>

Windows users will typically need to install Microsoft Windows SDK 7.1, whereas OS X users need to install XCode. Once a compiler is installed, it can be configured in Matlab with the command

```
mex -setup
```

If all else fails, ebFRET will revert to Matlab equivalents of the MEX functions. This means analysis will work as normal but will simply run slower.

---

<sup>4</sup><https://www.mathworks.com/matlabcentral/answers/62496>

<sup>5</sup><http://www.mathworks.com/support/bugreports/919688>