# FRET Data Analysis python

User documentation

## About FRET Data Analysis

FRET Data Analysis is a Python based software that allows the user to visualize his own Förster Resonance Energy Transfer (FRET) data and extract the relevant information of the bursts. The software is capable of analyzing and visualizing multi well FRET measurements.

## Compatibility

This software was developed and tested on macOS Monterey and Windows 10. Still it is expected to be executable on most systems. The software is not precompiled and does therefore require a python compiler. Using anaconda (https://anaconda.org) as the package manager his highly recommended.

­­­­­­­­

## Installing FRET Data Analysis

### Quick install

If you have a python compiler and Anaconda installed on your system the easiest way to install all packages is to just run the autoinstall.py file. Note that this script is in an early development state and my not compile. It is furthermore only compatible with macOS and Windows.

### Create an environment from .yml files

To help you creating your own anaconda environment we provide two environment files (Windows and Mac). To set up a FRET Data Analysis environment simply open the terminal and navigate to the requirements folder of the tool. After just enter:

$ conda env create --name pyBatTreeENV --file enviroment.yml

Your new environment has the name “pyBatTreeENV” (can be changed afterwards or while installing from the file). Note that instead of navigating to the file you can give as well a full path as the last statement in the bash command.

### List of required nonstandard packages

* Numpy (https://numpy.org/doc/stable/index.html)
* Pandas (https://pandas.pydata.org)
* Matplotlib (https://matplotlib.org)
* PyQt (https://www.qt.io)
* Scipy (https://scipy.org)
* Cv2 (<https://pypi.org/project/opencv-python/>)
* Joblib (https://joblib.readthedocs.io/en/latest/)

## Quick start guides

### pyBat

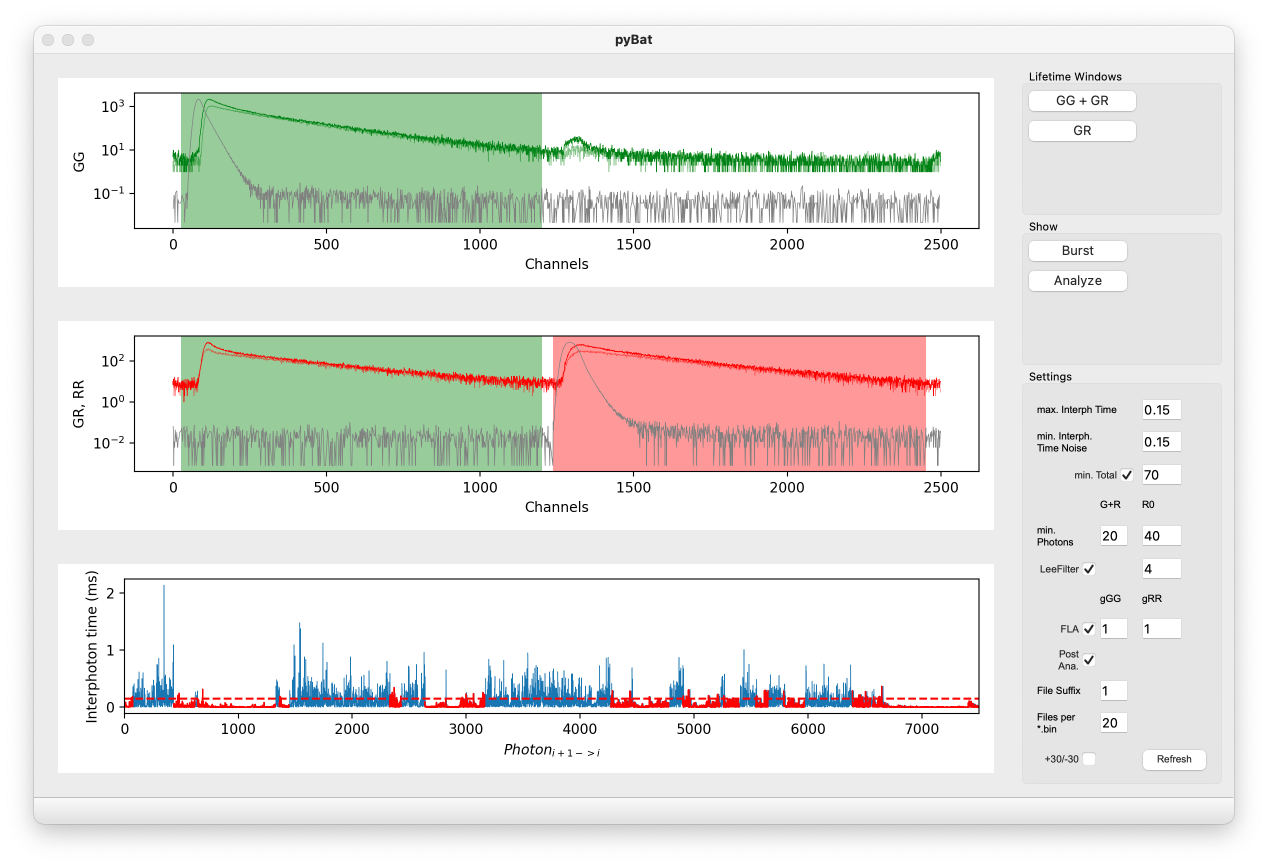


Figure 0.1 pyBat main window with data imported and ranges set. All settings are on default.

## Options

|  |  |  |
| --- | --- | --- |
| **Lifetime Windows** | | |
| 1 | DD + DA | Definition of the donor and acceptor emission channels after donor excitation |
| 2 | AA | Definition of the acceptor emission channel after acceptor excitation |
| **Show** | | |
| 3 | Raw data | Show fluorescence raw data of single loaded measurement file including photon time traces and 1ms binned intensity time traces |
| **Settings** | | |
| 4 | max. Interph. Time | Upper threshold of interphoton times for burst identification |
| 5 | min. Interph. Time Noise | Lower threshold of interphoton times for identification of background regions |
| 6 | min. Total | Minimal total number of consecutive photons satisfying the max. Interph. Time filter for burst identification |
| 7 | min. Photons  DD+DA and AA | Minimal number of consecutive DD+DA and AA photons satisfying the max. Interph. Time filter for burst identification |
| 8 | Lee Filter | Interphoton time smoothing filter blurring fluorescence fluctuations for burst identification |
| 9 | FLA | Fast lifetime analysis estimating the mean burst fluorescence lifetime of the donor and acceptor by subtraction of the mean delay time of the respective IRF from the average microtime of the donor/acceptor burst photons. If FLA is not selected donor and acceptor lifetimes are derived by maximum likelihood estimation using a convolution of the corresponding IRF and a single exponential distribution. |
| 10 | File Suffix | Start number of file enumeration |
| 11 | Files per \*.bin | Number of analyzed measurement files included in one Bin-file. |
| 12 | +30/-30 | Removal of 30 starting and ending photons of each burst to avoid diffusion artifacts. |
| 13 | Refresh | Refresh plotting windows using updated settings. |

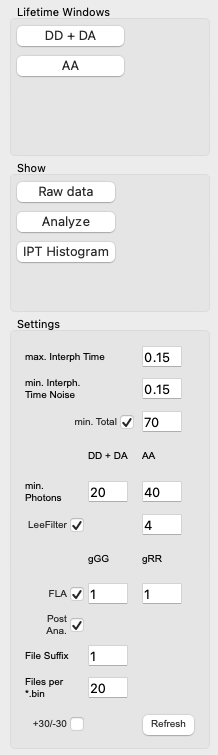


Figure 0.2 pyBat toolbar

## Visualization options

### Raw data

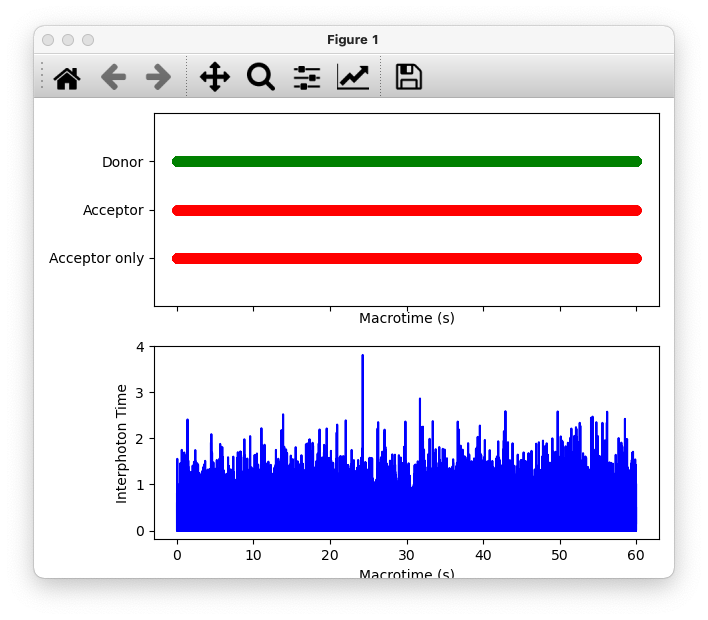


Figure 0.3: Showing macrotimes of RAW data. Top panel are the photons and bottom are Interphoton times. Figure can be zoomed in on both panels in parallel.

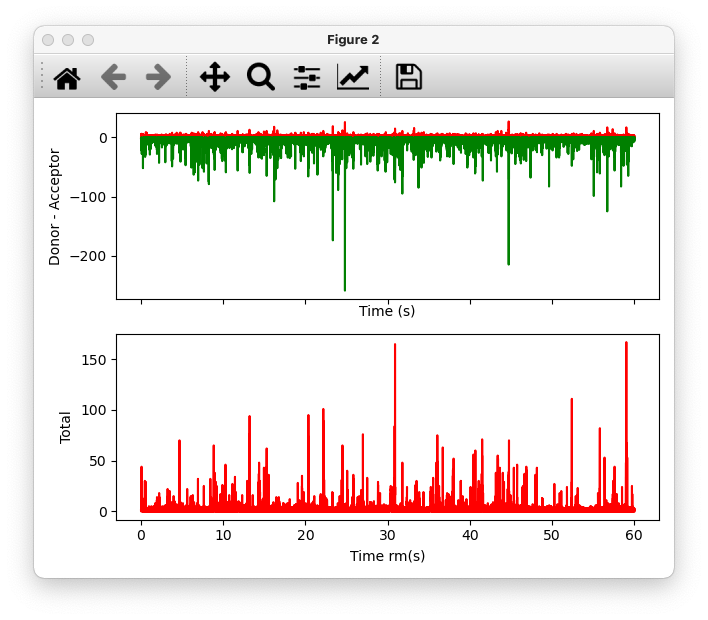


Figure 0.4: Showing Donor and Acceptor times (top panel) and Acceptor only (bottom panel).

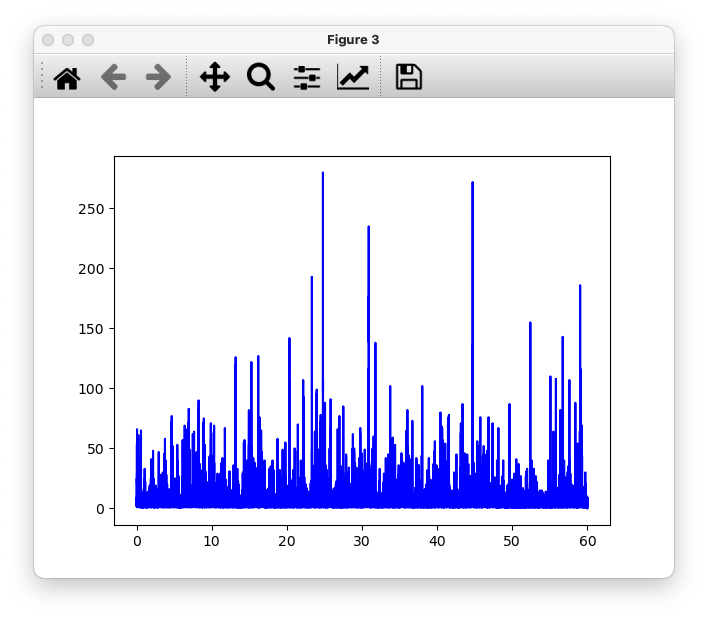
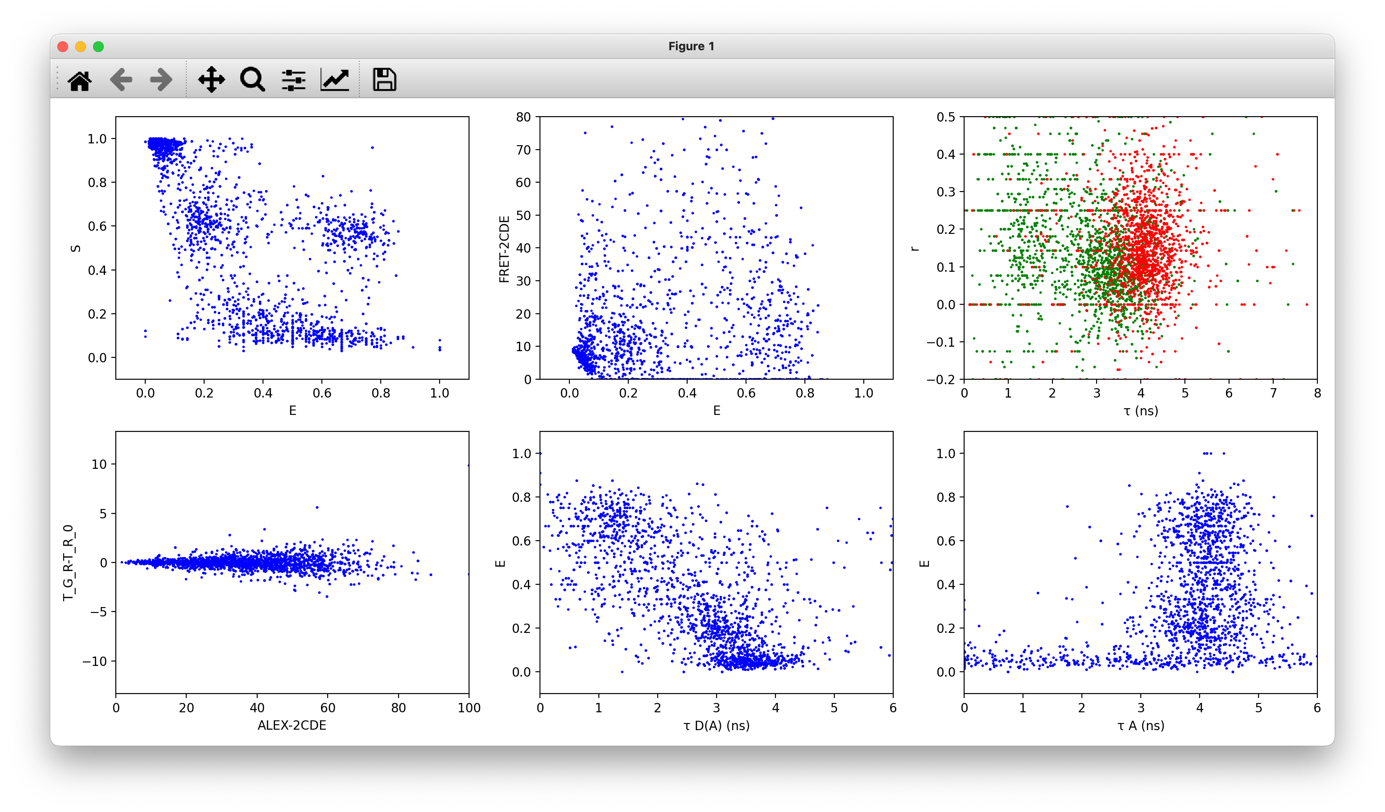


Figure 0.5: Showing The total vs. time.

### Analyze

Plots showing and updating while analyzing the folder. Note that this is not available when using the parallel version of Bat since the matplotlib commands do lead to errors when called from multiple threads within a parallel loop. We hope to fix this issue in future releases.



## Algorithms

### Raw data processing

The input data consists of the burst number, the channel number, the microtime and the macrotime. The micro- and the macro- time are given in the number of channel, which depends on the resolution of the instrument. The conversion factor from number of channel to microseconds must be specified once loading the data

#### Raw FRET efficiency

The raw FRET efficiency is the balance of resonant photons with respect to the total number of collected photons, :

#### Raw stoichiometry

The stoichiometry is the balance of photons obtained after excitation with green light, both green and resonant , with respect to the total number of collected photons, which includes also the photons collected after red excitation, :

#### Lifetime of the donor in the presence of the acceptor

The lifetime of the donor in the presence of the acceptor 𝜏 𝐷 ( 𝐴 ) recorded is the result of the convolution of the instrument response function (IRF) of the system and the li

#### Lifetime of the acceptor

Similarly, the lifetime of the acceptor fluorophore for a single burst is:

#### Burst Variance Analysis (BVA)

The BVA (Torella et al.) is a method to discern whether the heterogeneities in FRET results have their origin in dynamic processes or from a diversity of static states coexisting in the sample. It is based in the computation of the standard deviation of the FRET efficiency for packets of M photons. The calculation of the standard deviation is [2].

### Correction factors for the FRET efficiency and stoichiometry

The FRET efficiency, E, and the stoichiometry, S, can be corrected manually by the user by adjusting the factors α, β, and γ (Kudryavtsev et al.):

where , , and are the number of green, red and resonant photons respectively. The dimensionless factors α, β, and γ are set at 0, 0, and 1 by default, which correspond to the raw FRET efficiency.

The FRET efficiency in the case of the Burst Variance Analysis plot will not be modified by these adjustments.

## Requirements

From a hardware perspective, the software will run on most machines. However, the run time of an analysis will drastically vary depending on the system specs. After several test runs we conclude that a “slow system” with 2 Cores and 8Gb of memory is sufficient for the analysis of smaller sized datasets (12 measurements and total size of < 1Gb) in a reasonable time. For a full 96 well measurement (> 5Gb) we highly suggest a system with at least 16Gb of memory and 4 cores (8 threads) and the use of bat’s parallel version parBat.

## parBat

parBat is the parallel version of Bat aiming to decrease analysis time. It will automatically detect the number of available system threads. Each thread will handle the analysis of a single measurement folder representing the measurement of a single well. Performance will scale with the number of available system threads. In a first test we were able to analyze > 5Gb of measurement data in less than six hours. We hope to decrease the analysis time drastically in future releases.

## References

[1] Kudryavtsev, V., et al., “Combining MFD and PIE for Accurate Single-Pair Förster Resonance Energy Transfer Measurements”, ChemPhysChem 2012, 12, 1060-1078.

[2] Torella, J. P., et al., “Identifying molecular dynamics in single-molecule FRET experiments with burst variance analysis”, Biophysical Society 2011 Mar 16; 100(6):1568-77.