# 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 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/)

## Quick start guides

### pyBat

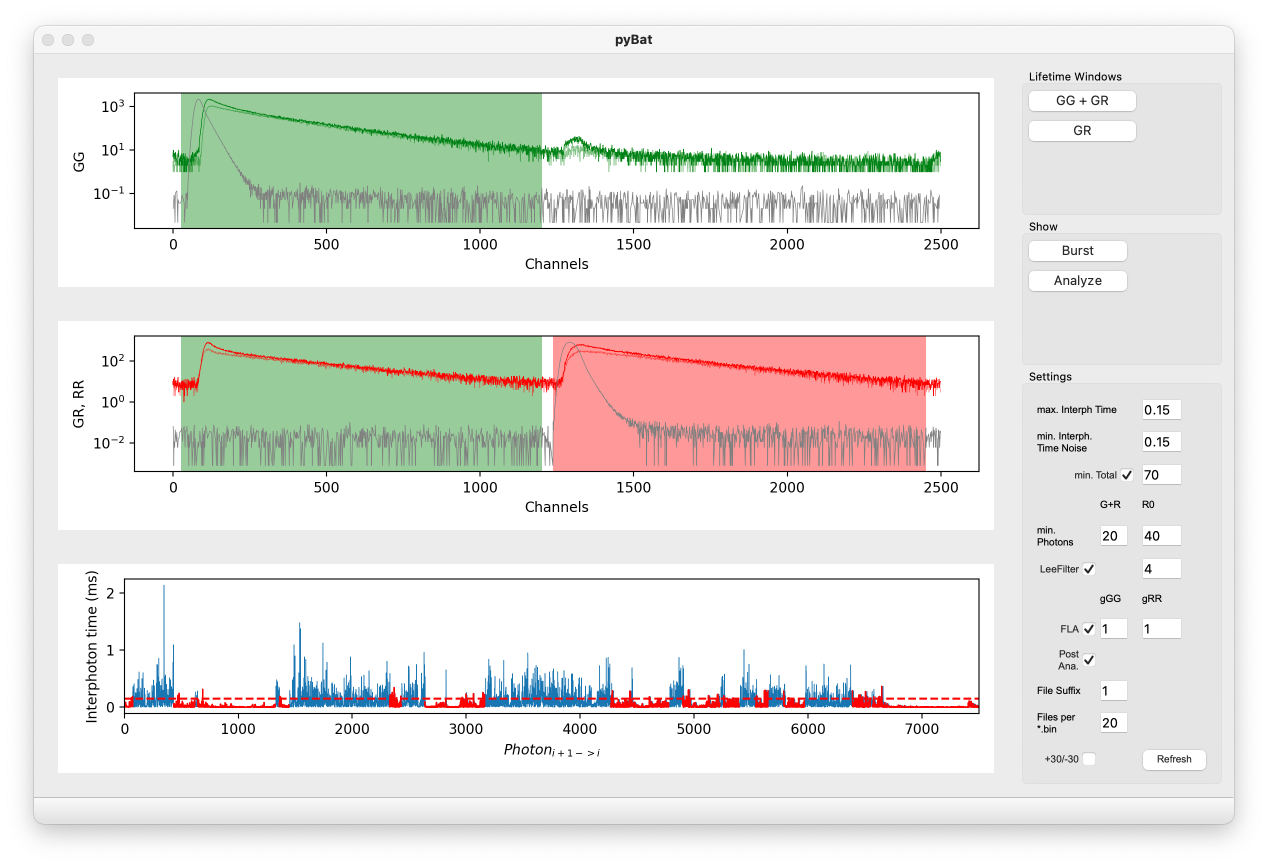


Figure 0.1 pyBat main window with data imported and ranges set. All settings are on default.

|  |  |  |
| --- | --- | --- |
| **Lifetime Windows** | | |
| 1 | GG + GR  DD + DA | Definition of the donor and acceptor emission channels after donor excitation |
| 2 | GR  AA | Definition of the acceptor emission channel after acceptor excitation |
| **Show** | | |
| 3 | Burst  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 identifictaion |
| 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  G+R and R0  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. |

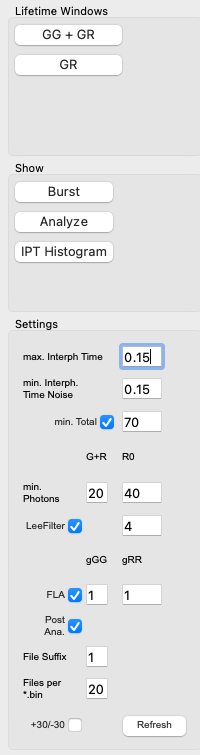


Figure 0.2 pyBat toolbar

## Visualization options

## Algorithms

## Requirements

## Abbreviations and acronyms list

## References