pyFRET: An Open-Source Python Library for Analysis of Single Molecule Fluorescence Data

Author1[1], Author2[2], Author3[3,*]

1 Rebecca R. Murphy Department of Chemistry, University of Cambridge, Cambridge, UK

2 Magnus Kjaegaard Interdisciplinary Nanoscience Center, Aarhus, Denmark

3 Sophie E. Jackson Department of Chemistry, University of Cambridge, Cambridge, UK

4 David Klenerman Department of Chemistry, University of Cambridge, Cambridge, UK

∗ E-mail: dk10012@cam.ac.uk

Abstract

Single molecule Förster resonance energy transfer (smFRET) is a powerful experimental technique for studying the properties of individual biological molecules in solution. In recent years, significant progress has been made in developing new data collection methods, which significantly enrich enhance the information that can be gained from a smFRET experiment. However, adoption of smFRET techniques has so far been limited by a reliance on custom built software for data processing, which is independently developed and maintained by each research group in the field.

Here, we present pyFRET, an open source python package for the analysis of data from single-molecule fluorescence experiments from freely diffusing biomolecules. The package provides methods for the complete analysis of a smFRET dataset, from burst selection and denoising, through data visualisation and model fitting. We provide support for both continuous excitation and alternating laser excitation (ALEX) data analysis, including burst search algorithms and stochastic denoising. We also implement the recently developed Recurrence Analysis of Single Particles (RASP) algorithm for kinetic analysis of fluorescent subppulations.

In this article, we demonstrate use of this package through analysis of a series of dual-labelled duplex DNAs, using both ALEX and continuous excitation data, to reproduce the characteristic sigmoidal FRET-eficiency curve. We also compare photon bursts from freely diffusing molecules with bursts from molecules flowed through the confocal volume, illustrating the effect of confocal dwell time on fluorescent emission.

pyFRET is available as a package downloadable from the Python Package Index (pyPI) under a three-clause BSD licence at `https://pypi.python.org/pypi/pyFRET/0.1.7.1`, together with links to extensive documentation and tutorials, including example usage and test data. Additional documentation including tutorials is hosted independently on ReadTheDocs, a dedicated documentation service (`https://readthedocs.org/projects/pyfret/`).

Author Summary

Introduction

Förster resonance energy transfer (FRET) [1] is a non-radiative energy transfer process that can occur between chromophoric molecules. The degree of energy transfer is dependent on the inter-fluorophore distance, allowing FRET to be used as a “molecular ruler” [1], to determine intramolecular distances. Since FRET was first used to measure the distance between two fluorescent dyes on individual molecules bound to a surface [2], single-molecule FRET (smFRET) has become a popular tool to investigate the structure and dynamics of biomolecules, both on a surface and diffusing freely in solution [3–5].

In a confocal smFRET experiment, biological molecules are labelled with two fluorescent dyes. The emission spectrum of the donor dye (D) is chosen to overlap with the excitation spectrum of the acceptor (A). When the donor and acceptor are sufficiently close in space, exciting the donor dye results in FRET

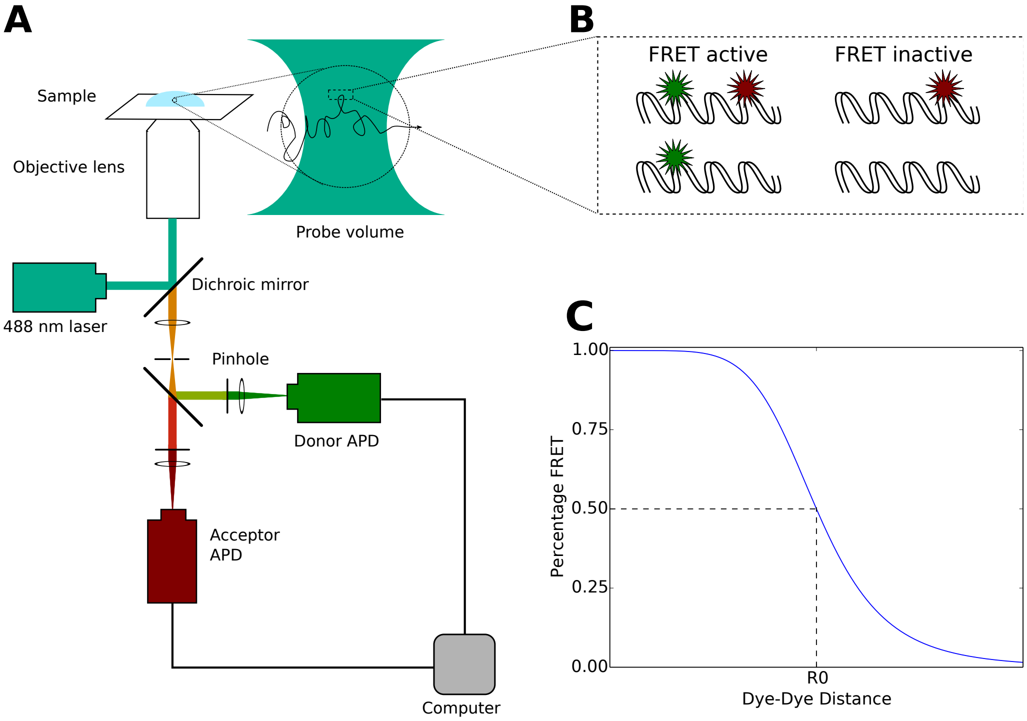

Figure 1. Instrumentation for a smFRET experiment. A) The confocal microscope, excitation and detection apparatus. B) Labelled molecules diffuse through the excitation volume. C) The characteristic sigmoidal dependence of FRET efficiency on dye-dye distance.

and fluorescent emission from the acceptor dye. The FRET efficeicny, E, which descrbes the proportion of excitation energy transferred from the donor to the acceptor, depends on the distance, r between the two dyes (Eq. 1) and R_0, the Förster distance, a dye dependent constant that describes the dye separation at which 50% energy transfer is achieved (Fig. **??** C)

$$E = \frac{1}{1 + (\frac{r}{R_0})^6} \tag{1}$$

Consequently, the distance between the two fluorophores can be determined from the ratio of donor and acceptor photons emitted during an excitation event (Eq. 3).

Experimentally, a collimated laser beam, is used to illuminate an extremely dilute solution of labelled molecules. When a labelled molecule diffuses through the laser beam, the donor dye is excited and photons are emitted from both donor and acceptor dyes. Emitted photons are collected through the objective and separated into donor and acceptor streams for collection and analysis (Fig. **??** A, B).

For a single fluorescent burst, the FRET efficiency, E, can be calculated as (Eq **??**):

$$E = \frac{n_A}{n_A + \gamma \cdot n_D} \tag{2}$$

for n_A and n_D detected acceptor and donor photons respectively and γ an experimentally determined instrument-dependent correction factor. During the course of a smFRET experiment, several thousand fluorescent bursts are collected and used to construct FRet Efficiency histograms. These histograms can be used to identify populations of fluorescent species [2], typically by fitting with multiple gaussian distributions.

Analysis of smFRET data involves several computational challenges. Firstly, photons emitted by a fluorescent molecle diffusing through the excitation volume must be identified against a noisy background. Secondly, identified bursts must be denoised, including removal of background auto-fluorescence and donor-acceptor crosstalk. Fluorescent bursts that are distorted by photobleaching or other photophysical artifacts should be identified and excluded. Multiple methods of burst selection and analysis have been developed and applied to the analysis of smFRET data [5–13]. However, software for analysis of smFRET data has thus far been developed on an ad hoc basis, with individual groups preparing and maintaining their own analysis scripts, leading to problems typical of research programming projects [14, 15].

Here we present pyFRET, an open-source python library, for the analysis of smFRET data. To our knowledge, this is the first open source software ever released by the smFRET research community. pyFRET is a small library that provides a toolkit facilitating all key steps in analysis of smFRET data: burst selection; cross-talk subtraction and burst denoising; data visualisation; and construction and simple fitting of FRET efficiency histograms. In providing this toolkit to the smFRET research community, we hope to facilitate the wider adoption of smFRET techniques in biological research as well as providing a framework for open communication about and sharing of data analsyis tools.

Design and Implementation

Code Layout and Design

pyFRET provides four key data structures (classes) for manipulation of smFRET data. The FRET data object describes two fluorescence channels, corresponding to time-bins containing photons collected from donor (the donor channel, D) and acceptor (the acceptor channel, A) fluorophores. The ALEX data object describes four fluorescence channels, corresponding to the four temporal states in a smFRET experiment using Alternating Laser Excitation (ALEX), namely the donor channel when the donor laser is switched on (D_D); the donor channel when the acceptor laser is switched on (D_A); the acceptor channel when the donor laser is on (A_D); and the acceptor channel when the acceptor laser is on (A_A). These data channels are implemented as numpy arrays, allowing efficient computations and selection operations.

Two similar classes are used for fluorescence bursts identified using the burst search algorithms. In addition to the donor and acceptor channels, the FRET bursts class type holds three additional arrays, giving the first and last bin of each burst, and the duration of each burst. These three new attributes are similarly present in the ALEX bursts object, in addition to the four fluorescent channels in the ALEX data object. In addition to the methods present for the simple FRET and ALEX objects, the burst data objects also implement methods to plot burst duration and to analyse recurrent bursts (RASP).

The data analysis workflow is illustrated in Fig. 2 Following initialization of data objects, background subtraction, event selection, cross-talk correction and calculation of the FRET efficiency can each be performed with a single line of code. Simple but high-quality figures can be generated in a single step.

Simple Event Selection and Denoising

FRET Data

In the most simple smFRET experiment, fluorescently labelled molecules are excited by a laser that will excite the donor dye; all photons reaching the detectors during data acquisition binned as they are received into time-bins of length proportional to the expected dwell-time of a molecule in the confocal volume (for freely diffusing molecules, a bin-time of 1 ms is typically used). Event selection then simply involves identifying time-bins that contain sufficient photons to meet a specified criterion. Two thresholding

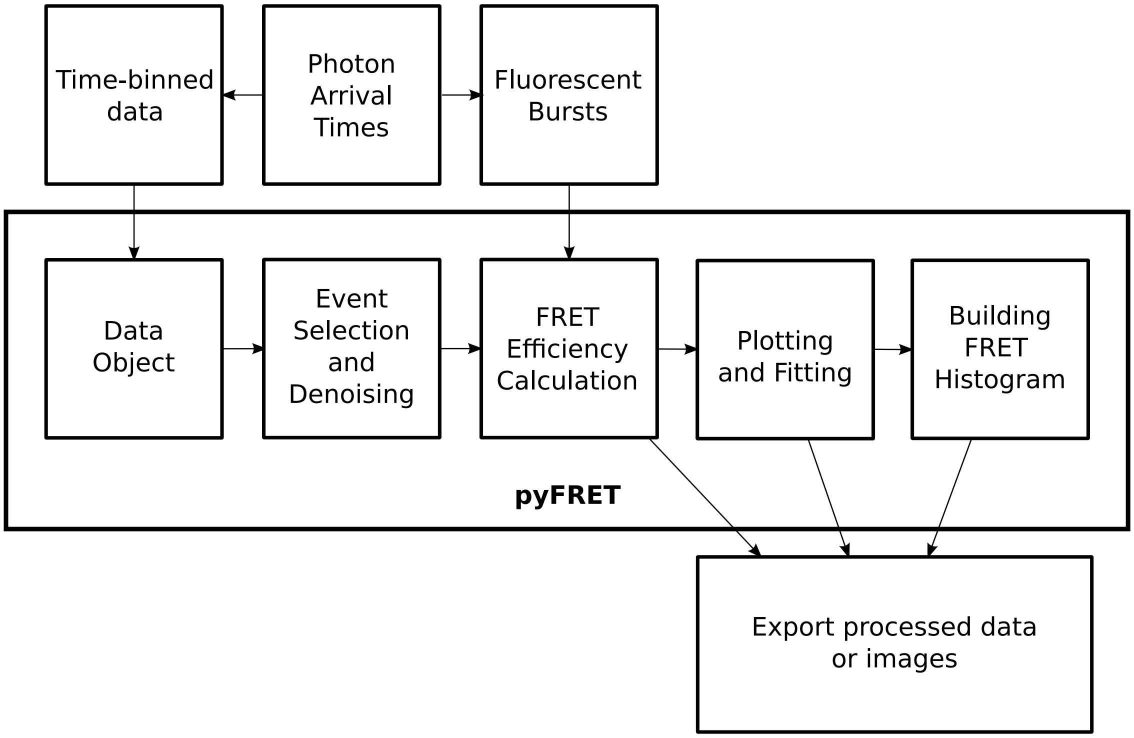

Figure 2. Typical workflow for data analysis using pyFRET.

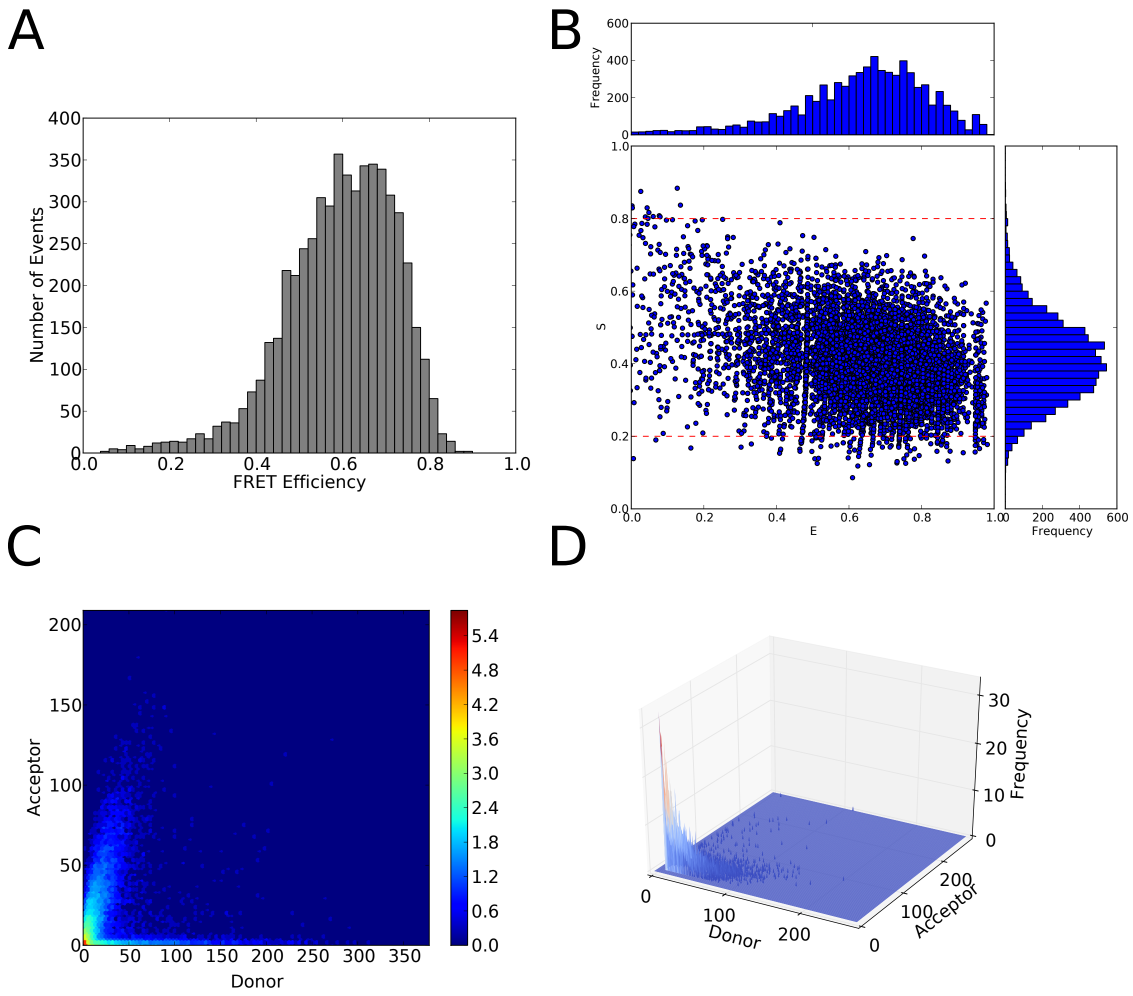

Figure 3. Figures made using pyFRET. A) A Proximity Ratio histogram. B) A scatter-plot of FRET efficiency and fluorophore stoichiometry from ALEX data. C) A heatmap of event frequencies. D) A 3D plot of event frequencies.

criterion are in common use. AND thresholding selects time bins for which $n_D > T_D$ AND $n_A > T_A$ for n_D and n_A photons in the donor and acceptor channels respectively, and T_D and T_A the donor and acceptor thresholds. In a similar manner, SUM thresholding considers the sum of photons observed in the donor and acceptor channels, selecting time bins for which $n_D + n_A > T$. These thresholding techniques can be implemented using a single call to a pyFRET function:

```
# Simple thresholding

# define thresholds
Td = 20 # donor threshold
Ta = 20 # acceptor threshold
T = 50  # combined threshold

# AND thresholding
data.threshold_AND(Td, Ta)

# SUM thresholding
data.threshold_SUM(T)
```

Following event selection, a simple method of denoising is to subtract from each selected event the average background autofluorescence observed in each channel and the average cross-talk between the two channels:

```
# Simple denoising

# removing autofluorescence
auto_donor = 0.5   # donor autofluorescence
auto_acceptor = 0.3 # acceptor autofluorescence
my_data.subtract_bckd(auto_donor, auto_acceptor)

# removing cross-talk
cross_DtoA = 0.05 # fractional cross-talk from donor to acceptor
cross_AtoD = 0.01 # fractional cross-talk from acceptor to donor
my_data.subtract_crosstalk(cross_DtoA, cross_AtoD)
```

This is the simplest method for event selection and denoising. However, it has several limitations. In particular, simple subtraction of constant values can lead to unphysical artifacts such as negative and fractional photon counts. Furthermore, the simple thresholding criteria for event selection are known to be biased [8], so can distort downstream data analysis.

ALEX Data

A more sophisticated smFRET experiment uses Alternating Laser Excitation (ALEX) during data acquisition. In this method, the diffuisng fluorescent molecules are subjected to excitation from both two lasers in rapid alternation [9]. One laser excites the donor fluorophore and the other can directly excite the acceptor fluorophore. The alternation of the laser excitation is fast on the timescale of molecular dwell-time in the confocal volume, allowing a single fluorescent molecule to receive multiple cycles of donor-acceptor direct excitation.

Instead of the two photon streams – donor and acceptor photons – observed in a simple smFRET experiment, in a confocal ALEX experiment, there are four streams: $F_{D_{ex}}^{D_{em}}$, $F_{D_{ex}}^{A_{em}}$, $F_{A_{ex}}^{D_{em}}$, $F_{A_{ex}}^{A_{em}}$. $F_{D_{ex}}^{D_{em}}$ and $F_{D_{ex}}^{A_{em}}$, respectively donor and acceptor emission during donor excitation, are analogous to the two original donor and acceptor photon streams in a smFRET experiment. $F_{A_{ex}}^{A_{em}}$ records acceptor emission during direct acceptor excitation, whilst $F_{A_{ex}}^{D_{em}}$ records donor emission during acceptor excitation. The presence of these extra channels provides additional information about the labelling state of molecules giving rise to fluorescent bursts, allowing exclusion of bursts from molecules lacking one of the FRET labels and bursts where one of the labels was bleached. Selection based on direct excitation of both fluorophores also removes the biases caused by simple AND or SUM thresholding.

pyFRET implements ALEX event selection as described in the original publication [16]. In brief, bursts are initially selected using a selection criterion based on the total number of photons emitted during donor and acceptor excitation: $F_{D_{ex}}^{D_{em}} + F_{D_{ex}}^{A_{em}} > T_D$ AND $F_{A_{ex}}^{A_{em}} > T_A$.

Following this initial event selection, a second selection step is performed, based on the ratio of photons emitted during donor and acceptor excitation periods. The photon stoichiometry, S is calculated for each burst as:

$$S = \frac{F_{D_{ex}}^{D_{em}} + F_{D_{ex}}^{A_{em}}}{F_{D_{ex}}^{D_{em}} + F_{D_{ex}}^{A_{em}} + F_{A_{ex}}^{A_{em}}} \tag{3}$$

Events for which the stoichiometry is either very close to one or very close to zero, indicating presence of only the donor or acceptor fluorophore respectively can be excluded using a second event selection criterion: $S_{min} < S < S_{max}$

Following event selection, remaining bursts can be corrected for photon leakage and direct excitation contributions. A two-dimensional scatter plot of FRET efficiency E and stoichiometery S is then produced, including one-dimensional histograms of both E and S.

pyFRET allows each of these steps to be performed separately, however they can also be combined into a single step combining event selection, denoising, FRET efficiency calculation and plotting:

```
# Simple ALEX analysis

g_factor = 0.95 # instrumental gamma factor
S_min = 0.2    # min accepted value of S
S_max = 0.8    # max accepted value of S
filepath = "path\to\my\file"
filename = "scatter_plot"
ALEX_data.scatter_hist(S_min, S_max, gamma=g_factor, save=True, filepath=filepath,
    imgname=filename, imgtype="png")
```

Example ALEX analysis scripts can be found in the supplementary information.

Burst Search Algorithms

Although using time-bins that are matched to the dwell-time of molecules in the confocal volume is simple, it is not ideal, as some bursts will be split over several bins, so may be counted as separate events, or not considered for analysis. More sophisticated event selection algorithms, typically called burst search algorithms [8], bin photons on a time-scale much shorter than the typical dwell time in the confocal volume and then scan the resultant photon stream for bursts of a specified duration and brightness. pyFRET implememts both All Photons Burst Search (ABBS) and a Dual Channel Burst Search (DCBS)algorithms for both ALEX data and for simple FRET data, as originally described [8].

In APBS burst search for FRET data, photons from both donor and acceptor channels are considered togehter. A burst is defined according to three constants: T, the averaging window; M, the minimum number of photons within window T; and L, the minimum total number of photons required for an identified burst to be retained. These three values are used in a two-step process for burst identification.

Firstly, “the start (respectively, the end) of a potential burst is detected when the number of photons in the averaging window of duration T is larger (respectively, smaller) than the minimum number of photons M.” [8]. In pyFRET, this intial search is performed using the convolve method from numpy [17] to provide a running sum across windows of T time-bins. Following initial burst identification, a burst is retained if it contains more than L photons [8].

The DCBS burst search is similar, but considers the donor and acceptor channels separately. For a burst to be accepted in DCBS, both channels must simultaneoulsy meet the running sum criterion, allowing exclusion of single colour bursts and bursts where one fluorophore bleaches.

The burst search algorithms implemented for ALEX data work in a similar manner. In the ALEX APBS method, bursts are identified by considering the total number of fluorescent photons $F_{total} = F_{D_{ex}}^{D_{em}} + F_{D_{ex}}^{A_{em}} + F_{A_{ex}}^{A_{em}}$ in each time bin. Bursts are identified where the running sum (calculated using the F_{total} photon stream) in the averaging window T exceeds M. The DCBS method considers donor excitation photons $F_{D_{ex}}^{D_{em}} + F_{D_{ex}}^{A_{em}}$ separately from photons emitted during direct acceptor excitation $F_{donor} = F_{A_{ex}}^{A_{em}}$, requiring that the running sum exceeds M for both F_{donor} and $F_{A_{ex}}^{A_{em}}$.

Example code for running a burst search algorithm is shown below:

```
# Burst search using FRET data

# required parameters
T = 50          # time window (bins)
M = 50          # first threshold
L = 60          # second threshold

# calling APBS algorithm
bursts_APBS = FRET_data.APBS(T, M, L)
```

Stochastic Denoising of Fluorescent Bursts

RASP: Recurrence Analysis of Single Particles

A recent innovation in confocal smFRET is Reccurrence Analysis of Single Particles (RASP) [18]. RASP uses the fact that fluorescent bursts ocurring within a small time window have a higher probility of being generated by the same molecule diffusing back through the confocal volume than from two independent fluorescent events, to access subpolulation interconversion kinetics.

RASP is a two-step process. First, initial bursts (b_1) with a FRET efficiency E_{b1} within some defined range $\Delta(E_{b1})$ are identified. Secondly, bursts (b_2) occurring within a time interval (called the recurrence interval) $T = (t_1, t_2)$ of b_1 are identified. Analysis of the distribution of FRET efficiencies in b_2, the population of recurrent bursts, provides information about the interconversion rate between subpopulations.

pyFRET implements RASP using array masking, to allow efficient selection of relevant bursts. RASP can be called in a single step from a FRET bursts or ALEX bursts object:

```
# RASP

# initial E range: 0.4 < E < 0.6
Emin = 0.4
Emax = 0.6

# Time interval for re-occurrence
# given in number of bins
Tmin = 1000
Tmax = 10000
```

```
# selcting re-ocurring bursts
recurrent_bursts = bursts_APBS.RASP(Emin, Emax, Tmin, Tmax)

# histogram of re-occurring bursts
recurrent_bursts.build_histogram(filepath, csvname, gamma=g_factor)
```

Compatibilities

pyFRET is written in Python. Both python 2 (v2.7) and python 3 (v3.3) are supported. pyFRET requires three further python libraries, namely numpy and scipy for data manipulation, and matplotlib for data visualisation. We recommend using the free Anaconda package bundle (`https://store.continuum.io/cshop/anaconda/`) to install these dependencies. Detailed installation instructions can be found in the pyFRET documentation (`http://pyfret.readthedocs.org/en/latest/tutorial.html#installing-pyfret`). pyFRET was written and tested in a Linux environment. However, it was written to be platform independent and has also been used successfully on both Apple and Windows machines.

The lack of Open Source software in the smFRET community has led to a proliferation of esoteric file-types used for data collection and storage. To make pyFRET as usable as possible for a wide range of smFRET researchers, we provide file parsers for simple .csv and .txt file formats, as well as our custom binary format. The pyFRET data structures can be initialised using simple python arrays of time-binned photons, for users whose file format is not currently supported. The tutorial and supplementary information provide example scripts for parsing common filetypes into pyFRET objects.

Experimental Methods

0.1 Analysis of DNA Duplexes

We tested the pyFRET library using DNA duplexes dual-labelled with Alexa Fluor 488 and Alexa Fluor 647. The duplex sequences and labelling sites can be found in the Supplementary Material. FRET data were collected for 15 minutes using continuous excitation at 488 nm and binned in intervals of 1 ms. ALEX data were collected for 15 minutes using alternating excitation at 488 and 640 nm, with a modulation rate of 0.1 ms, a dead-time of 0.1 μs and a delay compensation of 3 μs. ALEX data were then binned in intervals of 1 ms. The scripts and configuration files used to analyse these data using pyFRET can be found in the supplementary material.

0.2 Analysis of Blue Fluorescent Protein Equilibrium Unfolding

Results

Benchmarking Performance with DNA Duplexes

We tested the pyFRET library using DNA duplexes dual-labelled with Alexa Fluor 488 and Alexa Fluor 647.

Analysis of Equilibrium Unfolding of Blue Fluorescent Protein

Availability and Future Directions

pyFRET is available to download from PyPI under an open source MIT/BSD licence from the Python Package Index (`https://pypi.python.org/pypi/pyfret0.1.0`). Documentation can also be found here,

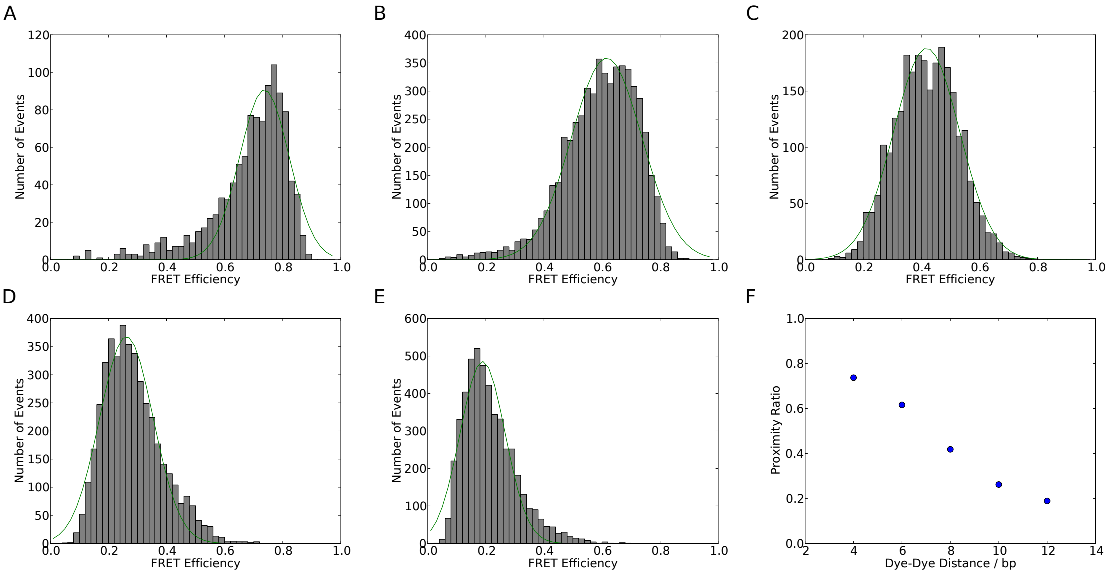

Figure 4. Analysis of FRET data from DNA duplexes using pyFRET. A - E: Fitted FRET histograms from DNA duplexes labelled with a dye-dye separation of 4, 6, 8, 10 and 12 base pairs respectively. F) Characteristic sigmoidal curve of FRET efficiency against dye-dye distance.

whilst a more extensive tutorial, including example scripts, can be found on our website (`http://rrm33.user.srcf.net/`) or in the Supplementary Information.

pyFRET currently provides basic tools for burst selection and denoising, based on simple thresholding and noise subtraction techniques. We are aware that more sophisticated methodologies exist and are currently working to produce and open source burst selection algorithm based on photon arrival times [8] as well as stochastic denoising algorithms [12]. We have also developed a novel analysis method based on Bayesian statistics [19], for which source code is available (`https://bitbucket.org/rebecca_roisin/fret-inference`) and which we intend to fold into the pyFRET library. We are also working to increase support for the wide variety of file formats that result from custom-built data collection hardware.

smFRET is a fast-developing and active research field and we are keen to support scientific progress through development of high-quality usable software. We are keen to work with others to enable their use of and contribution to the pyFRET library. We welcome requests for custom analysis requirements and are happy to support others who wish to contribute additional code to the pyFRET infrastucture.

Acknowledgments

RRM would like to thank BBSRC for PhD funding.

References

1. Forster T (1948) Zweischenmolekulare energiewanderung undfluoreszenz. Annalen der Physik 2: 55–75.

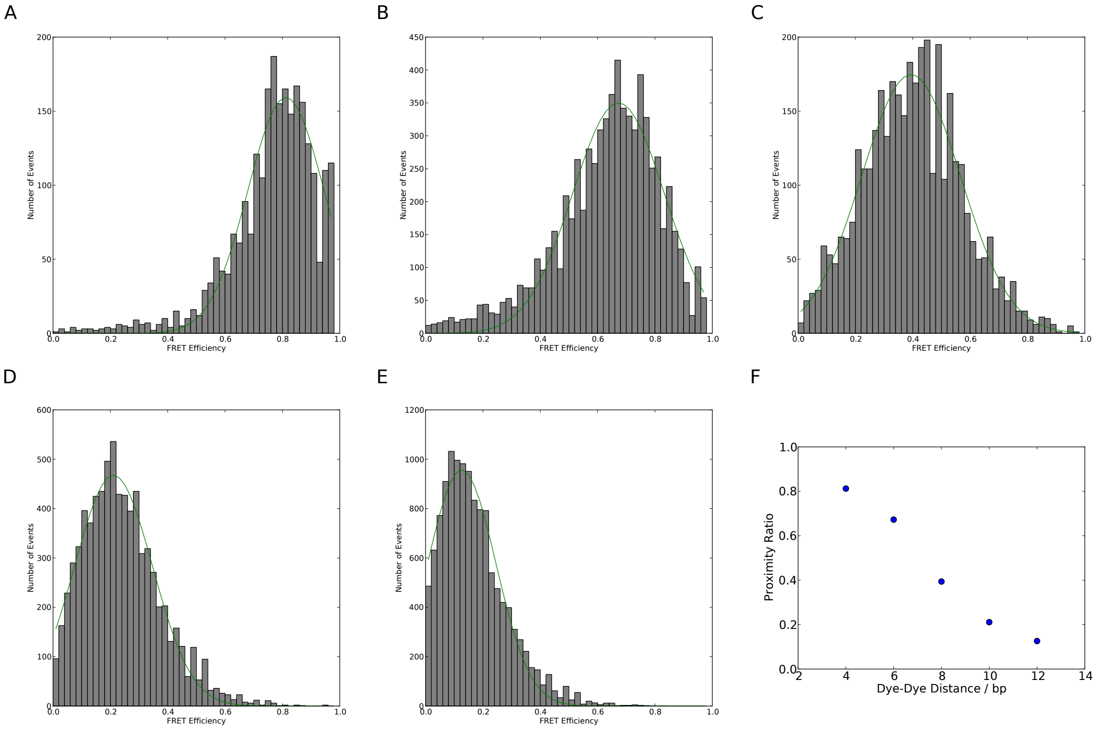

Figure 5. Analysis of ALEX data from DNA duplexes using pyALEX. A - E: Fitted FRET histograms from DNA duplexes labelled with a dye-dye separation of 4, 6, 8, 10 and 12 base pairs respectively. F) Characteristic sigmoidal curve of FRET efficiency against dye-dye distance.

2. Ha T, Enderle T, Ogletree DF, Chemla DS, Selvin PR, et al. (1996) Probing the interaction between two single molecules: Fluorescence resonance energy transfer between a single donor and a single acceptor. Proc Natl Acad Sci USA 93: 6264-6268.

3. Haran G (2003) Single-molecule fluorescence spectroscopy of biomolecular folding. J Phys: Condens Matter 15: R1291-R1317.

4. Schuler B, Lipman EA, Eaton WA (2002) Probing the free-energy surface for protein folding with single-molecule fluorescence spectroscopy. Nature 419: 743-747.

5. Weiss S (2000) Measuring conformational dynamics of biomolecules by single molecule fluorescence spectroscopy. Nat Struct Mol Biol 7: 724-729.

6. Deniz AA, Lawrence TA, Dahan M, Chemla DS, Schultz PS, et al. (2001) Ratiometric single-molecule studies of freely diffusing molecules. Annu Rev Phys Chem 52: 233–253.

7. Gell C, Brockwell D, Smith A (2006) Handbook of single molecule fluorescence. Oxford: Oxford University Press.

8. Nir E, Michalet X, Hamadani KM, Laurence TA, Neuhauser D, et al. (2006) Shot-noise limited single-molecule FRET histograms: Comparison between theory and experiments. J Phys Chem B 110: 22103-22124.

9. Kapanidis A, Laurence T, Lee N, Margeat E, Kong X, et al. (2005) Alternating-laser excitation of single molecules. Acc Chem Res 38: 532–533.

10. Muller BK, Zaychikov E, Brauchle C, Lamb DC (2005) Pulsed interleaved excitation. Biophys J 89: 3508-3522.

11. Doose S, Heilemann M, Michalet X, Weiss S, Kapanidis AN (2006) Periodic acceptor excitation spectroscopy of single molecules. Eur Biophys J 36: 669–674.

12. Kudryavtsev V, Sikor M, Kalinin S, Mokranjac D, Seidel CAM, et al. (2012) Combining mfd and pie for accurate single-pair frster resonance energy transfer measurements. ChemPhysChem 13: 1060–1078.

13. Eggeling C, Berger S, Brand L, Fries J, Schaffer J, et al. (2001) Data registration and selective single-molecule analysis using multi-parameter fluorescence detection. J Biotechnol 86: 163–180.

14. Wilson G (2006) Where's the real bottleneck in scientific computing? American Scientist 94: 5–6.

15. Merali Z (2010) Computational science: Error, why scientific programming does not compute. Nature 467: 775–777.

16. Lee NK, Kapanidis AN, Wang Y, Michalet X, Mukhopadhyay J, et al. (2006) Accurate fret measurements within single diffusing biomolecules using alternating-laser excitation. Biophys J 88: 2939–2953.

17. van der Walt S, Colbert SC, G V (2011) The numpy array: A structure for efficient numerical computation. Comput Sci Eng 13: 9–12.

18. Hoffmann A, Nettels D, Clark J, Borgia A, Radford SE, et al. (2011) Quantifying heterogeneity and conformational dynamics from single molecule fret of diffusing molecules: recurrence analysis of single particles (rasp). Phys Chem Chem Phys 13: 1857–1871.

19. Murphy RR, Danezis G, Horrocks MH, Jackson SE, Klenerman D (2014) Bayesian inference of accurate population sizes and fret efficiencies from single diffusing biomolecules. Anal Chem (submitted April 2014) : ?–?

Figure Legends

Tables