Journal of Visualized Experiments

Automated two-dimensional spatiotemporal analysis of mobile single-molecule FRET probes

--Manuscript Draft--

Article Type:	Invited Methods Collection - JoVE Produced Video
Manuscript Number:	JoVE63124R3
Full Title:	Automated two-dimensional spatiotemporal analysis of mobile single-molecule FRET probes
Corresponding Author:	Gerhard Schuetz AUSTRIA
Corresponding Author's Institution:	
Corresponding Author E-Mail:	schuetz@iap.tuwien.ac.at
Order of Authors:	Lukas Schrangl
	Janett Göhring
	Florian Kellner
	Johannes B. Huppa
	Gerhard J. Schütz
Additional Information:	
Question	**Response**
Please specify the section of the submitted manuscript.	Bioengineering
Please indicate whether this article will be Standard Access or Open Access.	Open Access ($3900)
Please indicate the **city, state/province, and country** where this article will be **filmed**. Please do not use abbreviations.	Vienna, Austria
Please confirm that you have read and agree to the terms and conditions of the author license agreement that applies below:	I agree to the Author License Agreement
Please confirm that you have read and agree to the terms and conditions of the video release that applies below:	I agree to the Video Release
Please provide any comments to the journal here.	

TITLE:

Automated Two-dimensional Spatiotemporal Analysis of Mobile Single-molecule FRET Probes

AUTHORS AND AFFILIATIONS:

Lukas Schrangl[1], Janett Göhring[2], Florian Kellner[2], Johannes B. Huppa[2], Gerhard J. Schütz[1]

[1]Institute of Applied Physics, TU Wien, Vienna, Austria
[2]Institute for Hygiene and Applied Immunology, Center for Pathophysiology, Infectiology and Immunology, Medical University of Vienna, Vienna, Austria

Email addresses of co-authors:

Lukas Schrangl	(lukas.schrangl@tuwien.ac.at)
Janett Göhring	(janett.goehring@meduniwien.ac.at)
Florian Kellner	(florian.kellner@meduniwien.ac.at)
Johannes B. Huppa	(johannes.huppa@meduniwien.ac.at)

Corresponding author:

Gerhard J. Schütz	(schuetz@iap.tuwien.ac.at)

SUMMARY:

This article presents a method for spatiotemporal analysis of mobile, single-molecule Förster resonance energy transfer (smFRET)-based probes using widefield fluorescence microscopy. The newly developed software toolkit allows the determination of smFRET time traces of moving probes, including the correct FRET efficiency and the molecular positions, as functions of time.

ABSTRACT:

Single-molecule Förster resonance energy transfer (smFRET) is a versatile technique reporting on distances in the sub-nanometer to nanometer range. It has been used in a wide range of biophysical and molecular biological experiments, including the measurement of molecular forces, characterization of conformational dynamics of biomolecules, observation of intracellular colocalization of proteins, and determination of receptor–ligand interaction times. In a widefield microscopy configuration, experiments are typically performed using surface-immobilized probes. Here, a method combining single-molecule tracking with alternating excitation (ALEX) smFRET experiments is presented, permitting the acquisition of smFRET time traces of surface-bound, yet mobile probes in plasma membranes or glass-supported lipid bilayers. For the analysis of recorded data, an automated, open-source software collection was developed supporting (i) the localization of fluorescent signals, (ii) single-particle tracking, (iii) determination of FRET-related quantities including correction factors, (iv) stringent verification of smFRET traces, and (v) intuitive presentation of the results. The generated data can conveniently be used as input for further exploration via specialized software, e.g., for the assessment of the diffusional behavior of probes or the investigation of FRET transitions.

INTRODUCTION:

Förster resonance energy transfer (FRET) has been a major driver in molecular biological and biophysical research, as it allows the investigation of processes at sub-nanometer resolution. As the efficiency of energy transfer between donor and acceptor fluorophores strongly depends on the inter-dye distance in the sub-nanometer to nanometer range, it has been effectively used as a spectroscopic ruler to explore static and dynamic conformation of biomolecules[1–4]. Additionally, the FRET phenomenon has been widely used for colocalization studies of membrane-associated and intracellular proteins on a bulk level[5,6]. In the last two decades, the method was adapted for monitoring smFRET events[7], which helped to substantially increase temporal and spatial resolution and resolved even rare subpopulations in heterogeneous samples. Equipped with these techniques, unique insights were gained into the dynamics of molecular machinery such as the transcript processing rate of RNA polymerase II[8], replication speed of DNA polymerases[9,10], nucleosome translocation rate[11], transcript splicing and stalling rate of assembled spliceosomes[12], the activity of ribosomal subpopulations[13], and the walking speed of kinesin motors[14], to name a few. Receptor–ligand interaction durations[15] and molecular forces[16] have been quantified.

Intensity-based smFRET studies typically rely on sensitized emission to measure FRET efficiency: a beam splitter in the emission path spatially separates light originating from donor and acceptor fluorophores upon donor excitation, allowing for the quantification of individual fluorescence intensities. The efficiency can subsequently be calculated as the fraction of photons emitted by the acceptor with respect to the total photon count[17]. In addition, acceptor excitation following donor excitation (ALEX) permits measurement of the stoichiometry of the FRET events, aiding in the discrimination between true low FRET signals from signals arising, e.g., from probes featuring a photobleached acceptor fluorophore[18].

Single-molecule FRET experiments are commonly carried out in one of two ways. First, a small region in the sample volume is illuminated using a confocal microscope. Single probe molecules in solution are excited when they happen to diffuse within the focal volume. With this technique, fast photon-counting detectors can be used, enabling sub-microsecond time resolution. Second, probes are specifically immobilized on surfaces and monitored via widefield microscopy, often using total internal reflection (TIR) configuration to minimize background fluorescence. Probe immobilization allows for much longer recording times than using the first approach. In addition, the larger field of view permits the monitoring of multiple probes in parallel. The need for a camera makes this method slow compared to the one described above. Time resolution is limited to the millisecond to second range.

If long time traces are required, e.g., for studying dynamic processes on a millisecond to second time scale, the first method is not applicable, as the fluorescence bursts are typically too short. The second approach fails whenever immobilization is not feasible, e.g., in live-cell experiments featuring probes diffusing within the cell membrane. Furthermore, it has been observed that biological model systems can vary their response dramatically depending on the mobility of the contacted surface[16].

While combined smFRET and single-particle tracking experiments recording mobile FRET probes have been performed in the past[19], there is no publicly available software for the evaluation of the data. This prompted the development of a new analysis platform, which allows for the determination of multiple properties of mobile fluorescent probes, including smFRET efficiency and stoichiometry, positions with sub-pixel accuracy, and fluorescence intensities as functions of time. Methods for filtering the resulting traces by examining stepwise bleaching behavior, nearest-neighbor distances, emission intensities, and other traits were established to exclusively choose correctly synthesized and functional single-probe molecules. The software also supports experimental and analytical techniques recently agreed upon in a multilaboratory study to produce reliable, quantitative smFRET data[17]. In particular, the implementation adheres to the validated procedures for the calculation of FRET efficiency and stoichiometry. Fluorescence intensities upon donor excitation in the donor emission channel I_{DD} and acceptor emission channel I_{DA} are used for the calculation of the apparent FRET efficiency E_{app} using Eq (**1**).

$$E_{\mathrm{app}} = I_{\mathrm{DA}}/(I_{\mathrm{DD}} + I_{\mathrm{DA}}) \qquad (\mathbf{1})$$

With the help of the fluorescence intensity in the acceptor emission channel upon acceptor excitation I_{AA}, the apparent stoichiometry is calculated using Eq (**2**).

$$S_{\mathrm{app}} = (I_{\mathrm{DD}} + I_{\mathrm{DA}})/(I_{\mathrm{DD}} + I_{\mathrm{DA}} + I_{\mathrm{AA}}) \qquad (\mathbf{2})$$

The FRET efficiency E and the stoichiometry S can be derived from E_{app} and S_{app} by considering four correction factors.

$$I_{\mathrm{DA}} \mapsto I_{\mathrm{DA}} - \alpha I_{\mathrm{DD}} - \delta I_{\mathrm{AA}}$$

Where α describes the leakage of donor fluorescence into the acceptor emission channel and can be determined using a sample containing only donor fluorophores or by analyzing parts of trajectories where the acceptor has been bleached. δ corrects for the direct excitation of the acceptor by the donor excitation light source and can be measured using a sample with only acceptor fluorophores or by analyzing parts of trajectories where the donor has been bleached.

$$I_{\mathrm{DD}} \mapsto \gamma I_{\mathrm{DD}}.$$

γ scales I_{DD} to rectify diverging detection efficiencies in donor and acceptor emission channels and different quantum efficiencies of the fluorophores. The factor can be computed by analyzing the increase in donor intensity upon acceptor bleaching in trajectories with high FRET efficiencies[20] or by studying a sample featuring multiple discrete FRET states.

$$I_{\mathrm{AA}} \mapsto I_{\mathrm{AA}}/\beta$$

β scales I_{AA} to correct for disparate efficiencies of donor and acceptor excitation. If γ was determined via acceptor bleaching analysis, β could be calculated from a sample of known $1:1$ donor-to-acceptor ratio[21]. Otherwise, the multi-state FRET sample also yields β.

Together, the corrections allow the calculation of the corrected FRET efficiency using Eq (**3**).

$$E = \frac{I_{\mathrm{DA}} - \alpha I_{\mathrm{DD}} - \delta I_{\mathrm{AA}}}{I_{\mathrm{DA}} - \alpha I_{\mathrm{DD}} - \delta I_{\mathrm{AA}} + \gamma I_{\mathrm{DD}}} \qquad (\mathbf{3})$$

and the corrected stoichiometry using Eq (**4**).

$$S = \frac{I_{\mathrm{DA}} - \alpha I_{\mathrm{DD}} - \delta I_{\mathrm{AA}} + \gamma I_{\mathrm{DD}}}{I_{\mathrm{DA}} - \alpha I_{\mathrm{DD}} - \delta I_{\mathrm{AA}} + \gamma I_{\mathrm{DD}} + I_{\mathrm{AA}}/\beta}. \qquad (\mathbf{4})$$

Ideally, the corrected stoichiometry for a 1:1 donor-to-acceptor ratio gives $S = 0.5$. In practice, a reduced signal-to-noise ratio produces a spread of the measured values of S, hampering the discrimination from donor-only signals ($S = 1$) and acceptor-only signals ($S = 0$). The resulting time traces can be used as input for a more detailed analysis of the single-molecule trajectories to obtain information such as spatiotemporal force profiles[16], the mobility of the single-molecule events[22], or transition kinetics between different states[1].

The following protocol describes experimental parameters and procedures for smFRET tracking experiments, as well as the working principle behind data analysis using the newly developed software suite. For the acquisition of experimental data, it is recommended to use a microscopy setup meeting the following requirements: i) capability of detecting the emission of single dye molecules; ii) widefield illumination: in particular for live-cell experiments, total internal reflection (TIR[23–25]) configuration is recommended; iii) spatial separation of emission light according to wavelength such that donor and acceptor fluorescence is projected onto different regions of the same camera chip[25] or different cameras; iv) modulation of light sources for donor and acceptor excitation with millisecond precision, e.g., using directly modulatable lasers or modulation via acousto-optic modulators. This permits stroboscopic illumination to minimize photobleaching of fluorophores as well as alternating excitation to determine stoichiometries. v) Output of one file per recorded image sequence in a format that can be read by the *PIMS* Python package[26]. In particular, multipage TIFF files are supported.

PROTOCOL:

1. **Software prerequisites**

1.1. Install the *miniconda* Python distribution[27] (minimum required Python version: 3.7).

1.2. Open an Anaconda prompt in the Windows Start menu, or open a terminal and execute **conda activate** if using Linux or macOS.

1.3. Enable the community-maintained **conda-forge** package repository[28] by executing the following commands:

```
conda config --add channels conda-forge
conda config --set channel_priority strict
conda update --all
```

1.4. Install the required Python packages by executing:

```
conda install opencv trackpy lmfit ipympl scikit-learn pyqt sdt-python jupyterlab
```

1.5. Become familiar with **JupyterLab**, the user interface of the analysis software (refer to the software documentation[29]).

1.6. Install the **git** version control system, which will be used later to download and update the analysis software. If using Linux, use the distribution's package management software to download and update. Otherwise, execute:

```
conda install git
```

1.7. Optionally install the sidecar Python package to display datasets after filtering steps during analysis:

```
conda install sidecar
```

2. **Measurement of samples** [Place **Figure 1** here]

2.1. When using an electron-multiplying charge-coupled device (EMCCD) camera, enable the EM gain to observe single-molecule signals at high signal-to-noise ratios (refer to the manufacturer's instructions).

2.2. Excitation sequence (see **Figure 1A** for further details).

2.2.1. Optionally record an image for segmentation to restrict data analysis to certain regions in the field of view. For instance, excite Fura-2-loaded cells using a 405 nm laser and capture their emission around 510 nm to evaluate only probes located in interfaces between cells and supported lipid bilayers (SLBs). Consequently, wait for time t_r to permit camera readout.

NOTE: On EMCCD cameras, t_r depends on the number of lines in the chosen region of interest (ROI). Therefore, choosing a small ROI can be advantageous because it reduces the delay between frames and the size of recorded data. Additionally, enabling frame transfer mode allows for further reduction of t_r.

2.2.2. Alternately excite donor and acceptor fluorophores repeatedly.

2.2.2.1. Excite the donor for an illumination time t_{ill} (5–10 ms is typically short enough to avoid motion blur) while also triggering the camera.

2.2.2.2. Wait for time t_{r} to permit camera readout.

2.2.2.3. Excite the acceptor for t_{ill} while triggering the camera.

2.2.2.4. Wait for a time t_{delay}.

NOTE: This must be longer than t_{r} to enable readout by the camera but can otherwise be chosen arbitrarily. It shall balance the requirements for time resolution and trace length.

2.2.2.5. Repeat steps 2.2.2.1–2.2.2.4. Choose the number of repeats to be large enough to ensure photobleaching of at least one fluorophore per probe within the field of view, which permits stepwise photobleaching analysis for discrimination of single-molecule signals from aggregates.

NOTE: Choosing appropriate t_{ill} and excitation laser intensities commonly requires some experimentation: The longer the illumination times and the higher the laser intensities, the better is the signal-to-noise ratio in the resulting images, but the shorter are the resulting time traces.

2.3. Record a sufficient number of movies for each sample.

3. **Additional measurements for the determination of correction factors**

3.1. Record a series of randomly placed fiducial markers visible in both emission channels for image registration (i.e., finding the transformation that maps coordinates of the donor emission channel onto the acceptor emission channel and vice versa). See **Figure 1B**.

NOTE: Image registration is performed by the software; see step 6.1.4.

3.2. Measure the intensity profile for both donor and acceptor excitation light sources for flatfield correction (i.e., correcting for inhomogeneous excitation across the field of view). To this end, prepare a sample featuring a high density of FRET probes and first acquire an image upon acceptor excitation, followed by photobleaching of the acceptor and subsequent recording of an image upon donor excitation. For increased stability, repeat several times in different sample regions. See **Figure 1C,D**. Alternatively, record a sample decorated with only the donor molecule and a second sample decorated with only the acceptor fluorophores.

NOTE: Flatfield correction is performed by the analysis software; see step 8.1.2.

3.3. Record a single-molecule sample (as in section 2) of a probe without an acceptor fluorophore to determine donor emission leaking into the acceptor channel.

NOTE: Donor leakage can also be computed from the actual probes' time traces after acceptor bleaching. If a sufficient number of such events is recorded, no additional measurement is necessary. Both options are supported by the analysis software; see **Supplemental Information**, section 3.15.

3.4. Acquire recordings of a probe without a donor fluorophore for the quantification of direct acceptor excitation by the donor excitation light source.

NOTE: Direct acceptor excitation can also be derived from the actual probes' time traces after donor bleaching. If a sufficient number of such events is recorded, no additional measurement is necessary. Both options are supported by the analysis software; see **Supplemental Information**, section 3.15.

3.5. Record a single-molecule sample featuring two distinct FRET efficiencies to correct for differing detection efficiencies of the donor and acceptor emission channels and differing quantum yields of the dyes.

NOTE: Such samples could be, for instance, Holliday junctions[1], which fluctuate between two conformations, or DNA rods that have FRET pairs attached at different, well-defined distances. If probes feature high and sufficiently constant FRET efficiencies, the correction can also be computed from acceptor bleaching events of probes' time traces, in which case no additional measurements are necessary. Both options are supported by the analysis software; see **Supplemental Information**, section 3.15.

4. Single-molecule localization algorithms

NOTE: Several analysis steps require single-molecule localization. Choose between a Gaussian fitting algorithm[30] and center-of-mass computation[31], depending on signal density, background, and signal-to-noise ratio.

4.1. To perform Gaussian fitting, choose the **3D-DAOSTORM**[30] algorithm via the respective user interfaces.

NOTE: **3D-DAOSTORM** is designed to distinguish even signals with overlapping point spread functions. While this is generally an advantage, it comes with a caveat: single, bright signals are occasionally identified as two adjacent ones, which can confuse the tracking algorithm and result in the detection of two short trajectories instead of a single long one.

4.2. Set the following parameters (for details, see the documentation of the **sdt-python** library[32], which provides the algorithm's implementation).

4.2.1. **Radius**: Set the initial σ value of the Gaussian fit function in pixels depending on the effective pixel size.

4.2.2. **Threshold**: Set a minimum amplitude (i.e., brightest pixel value, corrected for the estimated local background) for a local intensity maximum to be fit.

NOTE: The threshold is arguably the most important parameter. If set too low, noise may be considered a fluorescence signal, and bright signals may be fitted with two Gaussians. If set too high, dim signals are not fit.

4.2.3. **model**: Set to **2d** to fit circular Gaussians.

NOTE: The other models are not applicable to the smFRET data.

4.2.4. **find filter**: Apply a filter before finding local maxima to reduce noise, which is helpful in low signal-to-noise ratio situations. This can be i) **identity**: no filter; ii) **Crocker-Grier**: bandpass filter from Crocker-Grier algorithm[31,33]; or iii) **Gaussian**: a gaussian blur with σ set by the sigma parameter.

NOTE: For Crocker-Grier, the **feat. size** parameter should be roughly the radius of a point spread function in pixels. For Gaussian, fitting is performed using the unfiltered raw data.

4.2.5. **min. distance**: Fit two prospective signals separated by fewer than min. distance pixels by a single Gaussian.

NOTE: This can help in the aforementioned scenario where a bright signal is wrongly detected as two adjacent signals.

4.2.6. **size range**: Select minimum and maximum σ of the fits to remove detections from spurious signals due to noise.

4.2.6.1. Choose the **Crocker-Grier** algorithm via the respective user interfaces to perform center-of-mass computation (a refined algorithm[31] based on Crocker's and Grier's idea[33]).

NOTE: This algorithm is very robust even in low signal-to-noise scenarios and in dealing with signals featuring a range of intensities but cannot fit molecules with overlapping point spread functions precisely.

4.2.7. **radius**: Set the radius (in pixels) of a disk large enough to contain the whole point spread function.

4.2.8. **signal thresh.:** Set the minimum amplitude (brightest pixel above estimated background) for a local intensity maximum to be analyzed.

NOTE: If set too low, noise may be considered a fluorescence signal. If set too high, dim signals are not fit.

4.2.9. **mass thresh**.: Set the minimum total intensity (sum of background-corrected pixel values) of a signal to be analyzed.

NOTE: Same considerations as above apply.

5. **Software initialization**

5.1. Download analysis scripts. In an **Anaconda** prompt, navigate to a folder to save the analysis (using the **cd** command) and execute

git clone https://github.com/schuetzgroup/fret-analysis.git target folder

5.1.1. Replace the **target folder** with a descriptive name such as **2021-06-14_Force-FRET-experiment**.

NOTE: The analysis software will end up in this folder; ensure that this folder does not exist beforehand. It is recommended to download a copy of the analysis scripts for each experiment. This way, it is possible to revisit the analysis later, recall the parameters used, and make changes.

5.2. Copy **Jupyter notebooks** (**01. Tracking.ipynb**, **02. Analysis.ipynb, 03. Plots.ipynb**) into the newly created folder (henceforth referred to as the **root folder**). If this is the first time using the software, get them from the **notebooks** subfolder of the **root folder**.

NOTE: If similar datasets have already been analyzed, copying the notebooks from a previous experiment can be a convenient option, as parameters may have changed only slightly.

5.3. Start the JupyterLab server by executing the following command in the **Anaconda** prompt to open a web browser window displaying JupyterLab.

```
jupyter lab
```

NOTE: The browser is only the interface, while the process running in the **Anaconda** prompt is doing the actual work. As a consequence, closing the browser window has only minimal effect; the session can be restored by accessing **http://localhost:8888**. However, interrupting the JupyterLab process in the prompt or closing the prompt will terminate the analysis, leading to the loss of unsaved work.

5.4. In the JupyterLab browser window, use the left pane to navigate to the **root folder**. Double-click on **01. Tracking.ipynb** to launch the first notebook. After launching, look for a new tab to appear, which displays boxes, so-called cells, of Python code.

NOTE: All Jupyter notebooks feature comments describing the functionality of each code cell. Additionally, documentation for every method call can be displayed by placing the text cursor immediately before the opening **(** and press **Shift+Tab**.

5.5. See **Figure** 2 for an overview of the data analysis process. [Place **Figure 2** here]

NOTE: Sample data to try out the software can be downloaded from **https://github.com/schuetzgroup/fret-analysis/releases/tag/example_file**s.

6. **Localization, tracking, and fluorescence intensity analysis of single molecules** (**01. Tracking.ipynb**).

6.1. Use the **01. Tracking.ipynb Jupyter** notebook for the reliable analysis of fluorescence intensity values of single-molecule signals, which is tailored for the precise quantification, particularly of faint signals often occurring in FRET measurements (e.g., due to low donor signals at high FRET events and vice versa).

NOTE: To this end, direct integration of the pixel intensities in the raw data with correction for the local background is implemented. For screenshots of each analysis step and description of function call parameters, refer to the **Supplemental Information**.

6.1.1. Specify the illumination sequence to permit the selection of donor and acceptor excitation frames, as well as frames for image segmentation from recorded image sequences.

NOTE: As the software allows the processing of data recorded with arbitrary illumination protocols, it is necessary to indicate which frame in an image sequence was acquired while exciting what type of fluorophore; see **Supplemental Information**, section 1.2, step 3. The frame numbers of the original image sequence are preserved.

6.1.2. Describe and load datasets. Analyze multiple datasets at once, provided they were recorded using the same illumination settings. Assign an identifier and a pattern that matches the respective image sequence file names to each dataset. Additionally, define specific datasets for special purposes, such as recordings of fiducial markers for image registration, excitation light profiles for flatfield correction, and optionally donor-only and acceptor-only samples to determine correction factors.

6.1.3. Select emission channels in raw images if both channels were recorded using a single camera. For this, use the appropriate graphical widget to select the appropriate regions for donor and acceptor emission.

6.1.4. Localize fiducial markers in both emission channels and perform image registration. Use the provided user interface to find the appropriate parameters for the localization algorithm for both the donor and the acceptor emission channels. See section 4 for information about supported localization algorithms.

NOTE: Randomly distributed fiducial markers can be identified across emission channels by the spatial distribution of their nearest neighbors (**Figure 1B**). A custom implementation of the algorithm proposed for selective plane illumination microscopy[34] in the **sdt-python** library automatically matches the position of each marker in the donor emission channel with the position in the acceptor emission channel. A transformation T mapping the donor emission channel's coordinates to the acceptor emission channel's coordinates is found via a linear least-squares fit of an affine transformation to the markers' positions[35]. RANSAC is used to account for outliers, such as wrongly matched positions from the previous step.

6.1.5. Localize FRET probes independently upon donor and acceptor excitation in all frames and merge results into one table that contains the original frame number, 2-dimensional coordinates, and an identifier referring to the source image file.

NOTE: To this end, the software provides user interfaces to find appropriate options for the localization algorithm.

6.1.5.1. Localize FRET probes upon donor excitation in the sum of the images obtained from donor emission Im_{DD} and acceptor emission Im_{DA}, which hardly depend on the FRET efficiency. For information concerning the options for the localization algorithm, see section 4.

NOTE: Each sum image is calculated by transforming Im_{DD} using the transformation T previously obtained from image registration and added pixelwise to Im_{DA}.

6.1.5.2. Localize probes upon acceptor excitation in the acceptor emission channel Im_{AA} (see section 4 for details about the localization algorithms).

6.1.6. Perform tracking and fluorescence intensity measurement. [Place **Figure 3** here]

6.1.6.1. Choose appropriate options for the **trackpy**[36] algorithm used to link FRET probe localizations into trajectories. In particular, set the maximum search distance from one frame to the next and the number of consecutive frames for which a signal may go undetected, which can occur due to bleaching or missed localizations.

NOTE: These gaps are filled via interpolation between preceding and ensuing positions. These interpolated positions are marked and only used later for reading out intensity values but not for diffusion analysis. Tracks are analyzed in the coordinate system of the acceptor emission channel. For fluorescence intensity analysis (step 6.1.6.2), tracks are additionally transformed to the coordinate system of the donor emission channel using the inverse transformation T^{-1} obtained via image registration (see step 6.1.4).

6.1.6.2. Select options for the fluorescence intensity calculation algorithm (see **Figure 3A** for details). Specify i) the radius of a disk that, when centered on a signal's position, contains all pixels affected by that signal and ii) the width of a ring around each disk used to determine the local background.

NOTE: To reduce noise in the obtained intensity measurements, a Gaussian blur with a standard deviation of 1 pixel is applied to the images (**Figure 3B,C**).

6.1.7. Use the analysis software functionality to process auxiliary image data from image sequences.

6.1.7.1. Extract additional images recorded to facilitate segmentation (see step 2.2.1, marked by **s** in the excitation sequence (see **Supplemental Information**, section 1.2, step 3).

6.1.7.2. Determine donor and acceptor excitation light profiles across the field of view from images recorded on a densely labeled sample (see step 3.2).

NOTE: The pixel-by-pixel mean is calculated from the images to compute the light profiles. The camera baseline is subtracted. Images are blurred using a Gaussian filter to reduce effects due to sample impurities. Finally, the resulting images are divided pixelwise by their maximum value to get the profile $p(x,y)$ mapping coordinates onto the interval $[0,1)$.

7. **Visualization of FRET trajectories** (optional)

7.1. Use the **inspector** application to display single-molecule tracks in raw image data and the corresponding fluorescence intensities and apparent FRET efficiencies and stoichiometries.

NOTE: This is a valuable tool to assess the validity of chosen parameters and manually accept or reject individual time traces. See **Supplemental Information** for a screenshot and detailed usage information.

8. **Analysis and filtering of single-molecule data (02. Analysis.ipynb)**

8.1. Use the **02. Analysis.ipynb Jupyter** notebook for analysis and filtering of the single-molecule data obtained via the **01. Tracking.ipynb** notebook. See the steps below for a typical analysis pipeline.

NOTE: Different scientific questions and experimental designs may require adjustments of the settings. The use of Jupyter notebooks permits easy adaption by omitting, rearranging, and amending analysis steps. For screenshots of each analysis step and description of function call parameters, refer to **Supplemental Information**.

8.1.1. Perform initial filtering steps.

8.1.1.1. Discard signals with overlapping point spread functions as it is difficult to determine their fluorescence intensities reliably.

8.1.1.2. In the case of inhomogeneous illumination, accept only signals located in well-illuminated regions within the field of view to ensure a good signal-to-noise ratio.

8.1.1.3. If studying intramolecular FRET, restrict the analysis to those trajectories present from the beginning of the image sequence to ensure that all bleaching steps are recorded and can be properly evaluated later during the stepwise photobleaching analysis.

NOTE: When performing experiments with intermolecular FRET probes, wherein donor and acceptor fluorophores are not part of a preformed complex, it may not be feasible to restrict the analysis to initially present trajectories.

8.1.2. Execute flatfield correction, which uses the excitation light source profiles $p(x, y)$ obtained in step 6.1.7.2 to reverse the position-dependent fluorescence intensity variations caused by inhomogeneous illumination.

NOTE: The fluorescence intensity $I(x, y)$ of a probe at the position (x, y) is corrected via $I(x, y) \rightarrow I\,(x, y)/p\,(x, y)$; see **Figure 1C,D**.

8.1.3. Compute the apparent FRET efficiency E_{app} (i.e., the fraction of energy transmitted from the donor fluorophore to the acceptor fluorophore) and the apparent stoichiometry S_{app} (i.e., the number of donor fluorophores divided by the total number of fluorophores within a diffraction-limited spot).

NOTE: By plotting E vs. S for each data point, it is possible to distinguish alterations in the measured FRET efficiencies due to change in donor—acceptor distance from alterations due to changes in the stoichiometry[18]. This allows for the differentiation between $E = 0$ due to dye separation from $E = 0$ due to the absence of an active acceptor. E–S plots are used throughout the analysis as a tool for quality assessment; see **Figure 4** as an example.

8.1.4. Perform stepwise analysis of photobleaching for the discrimination between single-molecular probes and aggregates. Choose to accept one of the following options.

NOTE: To this end, the analysis software applies a custom implementation[32] of the changepoint detection algorithm PELT[37] separately to the fluorescence intensity upon donor excitation ($I_{\text{DD}} + I_{\text{DA}}$) and acceptor excitation (I_{AA}).

8.1.4.1. Choose option 1, wherein the acceptor fluorophore bleaches in a single step while the donor shows no partial bleaching (i.e., there is no bleaching step to non-zero intensity).

NOTE: This option further rejects trajectories where the donor bleaches before the acceptor in a single step. Option 1 is the preferred choice in case of high acceptor photobleaching rates.

8.1.4.2. Choose option 2, wherein the donor bleaches in a single step while there is no partial acceptor bleaching.

NOTE: This option further rejects trajectories where the donor bleaches after the acceptor in a single step. Option 2 is the preferred choice in case of high donor photobleaching rates.

8.1.4.3. Choose option 3, wherein either fluorophore bleaches in a single step while the other does not partially bleach.

NOTE: Option 3 gives higher flexibility than options 1 and 2 and would be the suggested preference for data analysis.

8.1.4.4. Choose option 4, wherein donor and acceptor fluorophores show single-step photobleaching or no photobleaching at all.

NOTE: Option 4 is preferred in case of low photobleaching rates.

8.1.5. Calculate the correction factors for donor emission leakage into the acceptor channel α, direct acceptor excitation δ, detection efficiencies γ, and excitation efficiencies β[17].

8.1.6. Use the correction factors to calculate the FRET efficiency E from the apparent efficiency E_{app} and the stoichiometry S from the apparent stoichiometry S_{app}

8.1.7. Perform further filtering steps. Select only data points from before the first bleaching event in each trajectory. Additionally, accept only trajectories with at least 75% of data points satisfying $0.35 < S < 0.6$ to restrict the analysis to single-molecule probes (numbers are adjustable).

NOTE: The upper and lower bounds for S should be chosen according to the spread of the population of interest versus the populations to be excluded from the analysis (e.g., donor-only and acceptor-only populations). Based on experience, $0.35 < S < 0.6$ turned out to be a good choice for many experimental situations.

8.1.8. Perform image segmentation via global or adaptive thresholding methods[35] on the appropriate auxiliary images (see steps 2.2.1 and 6.1.7) to restrict the analysis to distinct regions within the field of view.

NOTE: This permits, for instance, exclusive evaluation of probes located in a cell—SLB interface or on a patterned structure.

9. **Plotting of results and further analysis (03. Plot.ipynb)**

NOTE: Refer to **Supplemental Information** for screenshots of the Jupyter notebook and description of function call parameters.

9.1. Create E–S plots to verify that signals of incorrect stoichiometry have been correctly identified and removed.

9.2. Plot histograms of FRET efficiencies to provide a well-established overview of the FRET efficiency distributions. Group the histograms for convenient comparison of results from different experiments.

9.3. Evaluate the data further (e.g., diffusion analysis, conversion of FRET efficiencies to forces in experiments using molecular force sensors or transition analysis) within the notebook taking advantage of scientific Python libraries.

NOTE: Data can also be exported in many file formats as input to other analysis software.

REPRESENTATIVE RESULTS:

A variety of low- and high-level information can be extracted from smFRET tracks depending on the scientific question of the experiment. Here, examples of analysis pipelines with analog and digital probes are presented: a peptide-based molecular force sensor[16] and a DNA probe with stochastic switching of its conformation[38], respectively. Refer to **Figure 5** for the design and working principle of these probes.

After the localization and tracking algorithms have been executed as described in the protocol, the package offers multiple data visualization tools to optimize the initial parameters and subsequent filter steps: (i) visualization of individual smFRET events, (ii) optional image segmentation to analyze data in certain regions of interests, (iii) monitoring of filter steps via FRET efficiency vs. stoichiometry (E–S) plots. The visualization of the single-molecule data is presented in **Figure 6**.

Finally, the filtered FRET events are represented by an E–S plot and a FRET efficiency histogram (**Figure 4**). The E–S plot is a useful tool for optimizing the aforementioned filtering steps and investigating the final result. Partially bleached or incompletely labeled FRET sensors can be excluded by their stoichiometry value. Mobility parameters can be investigated by plotting an individual trajectory path in an $x—y$ plot (**Figure 6**) or a mean square displacement (MSD) plot (**Figure 4**). The first method is especially useful for discriminating mobile from immobilized events, while the latter is used to calculate the diffusion coefficient.

FIGURE AND TABLE LEGENDS:

Figure 1: Image Acquisition. (**A**) Excitation sequence. After recording an optional image of a dye-loaded cell using the 405 nm laser, donor and acceptor are excited alternately and repeatedly for illumination time t_{ill} using 532 nm and 640 nm lasers, respectively. The time t_{r} between donor and acceptor excitation must be long enough to allow for image readout by the

camera. The delay time t_{delay} can be used to adjust the acquisition frame rate and, therefore, the observation time span before photobleaching. This panel is modified from [16]. (**B**) Fiducial markers are used for the calculation of the coordinate transforms between the two emission channels. Matching fiducials are indicated by color. Several shifted images should be recorded to ensure that the whole field of view is covered. (**C**) Laser profiles for flatfield correction are recorded using a densely labeled sample. The acceptor profile is recorded and photobleached, followed by acquisition of the donor profile. Multiple images should be taken at different sample regions, averaged, and smoothed to mitigate the influence of sample imperfections (e.g., the bright spot in the center-top of the image). (**D**) Flatfield correction map $p(x, y)$ calculated from 20 laser profile recorded as described in **C**. Abbreviations: FRET = Förster resonance energy transfer; Im_{DD} = donor emission image upon donor excitation; Im_{DA} = acceptor emission image upon donor excitation; Im_{AA} = acceptor emission image upon donor excitation. Scale bars = 5 µm.

Figure 2: Overview of a typical analysis pipeline. Note that filtering steps are subject to adaptation according to the experimental design. This figure is modified from [16]. Abbreviation: FRET = Förster resonance energy transfer.

Figure 3: Single-molecule intensity measurement. (**A**) For a fluorophore located at the orange pixel, its uncorrected intensity I_{uncorr} is determined by summing up all pixels' intensities within a disk (yellow and orange pixels) large enough to cover all pixels affected by the signal: $I_{\mathrm{uncorr}} = \sum_{px\in\mathrm{disk}} I_{px}$. The local background b is computed as the mean of the pixels in a ring (blue pixels) around the disk: $b = \sum_{px\in\mathrm{ring}} I_{px}/n_{\mathrm{ring}}$, where n_{ring} is the number of pixels in the ring. The fluorescence intensity I is the result of subtracting the background from the uncorrected intensity, $I = I_{\mathrm{uncorr}} - b \times n_{\mathrm{disk}}$, where n_{disk} is the number of pixels in the disk. The circle radius is specified via the **feat_radius** parameter of the tracking method. The width of the ring is given by the **bg_frame** parameter. If the point spread function of one signal overlaps with the background ring of another (bottom panel), the affected pixels (red) are excluded from local background analysis. If two point spread functions overlap, fluorescence intensities cannot be calculated reliably and are therefore discarded. (**B, C**) Simulations show that applying a Gaussian blur with a standard deviation of 1 pixel improves the signal-to-noise ratio up to a factor of close to 2 at low fluorescence intensities (**B**) and introduces hardly any error (slight underestimation of less than 1%, (**C**))[16]. Moreover, the relative error (i.e., $(I_{\mathrm{meas}} - I_{\mathrm{truth}})/I_{\mathrm{truth}}$, where I_{truth} is the ground truth and I_{meas} is the outcome of the analysis) is constant over the whole intensity range and therefore cancels out for ratiometric quantities such as FRET efficiencies and stoichiometries. All plots are based on previously published work[16]. Abbreviations: SNR = signal-to-noise ratio; FRET = Förster resonance energy transfer.

Figure 4: Exemplary output. (**A**) The FRET efficiency is plotted versus stoichiometry (E–S plot) for a population of the molecular force sensor (left panel) decorating a glass-supported lipid bilayer and strained by a T cell. Only one population cloud is visible. The respective histogram of FRET efficiencies exemplifies the difference between a force sensor population in presence and absence of cells (middle panel). No shift to lower FRET efficiencies of the sensor population in presence of T cells can be observed, indicating little to no force-dependent stretching of the

sensor module. The MSD plot of these experimental conditions confirms that the force sensor population beneath a T cell moves considerably slower than their unbound counterparts (right panel). (**B**) The same analysis was performed with Holliday junction DNA sensor decorating a glass-supported fluid lipid bilayer. The E–S plot clearly shows two populations, which are also apparent in the FRET efficiency histogram. The MSD plot indicates the presence of one fast-moving sensor population. Abbreviations: FRET = Förster resonance energy transfer; MSD = mean square displacement.

Figure 5: Design and working principle of intramolecular FRET probes. (**A**) Analog peptide sensor for quantification of mechanical molecular forces. The donor and acceptor fluorophores are covalently attached to either end of the peptide backbone. The sensor module is site-specifically attached to a specific ligand, which in turn binds a cell-resident surface receptor of interest (here, an antibody fragment specifically recognizing the beta chain of the T cell receptor). Upon receptor–ligand binding, force is exerted, and the sensor module extends and eventually recoils after bond cleavage. This panel is modified from [16]. (**B**) Digital DNA sensor for quantification of FRET transitions. The FRET sensor is composed of four DNA strands forming a Holliday junction. The donor and acceptor fluorophore are covalently attached to two strands. Holliday junctions frequently switch their conformation depending on the surrounding buffer conditions. The stochastic switching of these conformations can be monitored by quantifying the FRET efficiency of individual probes. Abbreviations: TCR = T cell receptor; FRET = Förster resonance energy transfer.

Figure 6: Examples of localization and tracking of FRET probes. (**A**) The FRET efficiency and stoichiometry of individual events are calculated by quantifying the intensity of the donor fluorophore upon donor excitation (D → D), the acceptor fluorophore upon donor excitation (D → A), and the acceptor fluorophore upon acceptor excitation (A → A). Nearest neighbor filtering prevents bias by overlapping point spread functions of close emitters. Image segmentation allows the user to choose certain smFRET events localized within an area of interest (e.g., a cell or a micropattern). As an example of image segmentation, T cells were stained with Fura-2 (displayed on the left) and subjected to adaptive thresholding to identify the cell edges (orange dotted line). Scale bars = 5 µm. (**B**) smFRET trajectories using the molecular force sensor. Individual trajectories can be plotted in the x–y plane, visualizing their diffusion behavior and localization (left panel). Furthermore, each trajectory's intensities can be plotted over time to identify FRET transitions or bleaching steps (middle panel shows the red trajectory from the left panel). The resulting FRET efficiency and stoichiometry can be visualized similarly (right panel). (**C**) smFRET trajectories using the Holliday junction DNA sensor. HBSS + 12 mM $MgCl_2$ was used as a buffer during the measurements. Apart from the apparent acceptor bleaching step near the sequence end of these examples, the frequency of FRET transitions for each sensor can be determined. The Holliday junctions switch their conformation with a high frequency, whereas the molecular force sensor does not exhibit FRET transitions. This information makes it possible to adjust the experimental conditions, such as the delay between the frames, to increase or reduce the number of observed transitions. Abbreviations: FRET = Förster resonance energy transfer; smFRET = single-molecule FRET; HBSS = Hank's balanced salt solution.

Supplemental Information: Localization and tracking of single molecules (01. Tracking.ipynb).

DISCUSSION:

This article details a pipeline for the automated recordings and quantitative analysis of smFRET data originating from mobile yet surface-tethered probe molecules. It complements the two predominant approaches to smFRET experiments, involving either surface-immobilized probes or probes diffusing in solution into and out of a confocal excitation volume[17]. It provides the correct FRET efficiency and the molecular positions as a function of time. It can therefore be used as input for specialized analysis programs, e.g., to quantify transition kinetics[1], FRET histograms[39], or two-dimensional diffusion[22].

The software is released under a free and open-source license approved by the Open Source Initiative that grants the user the perpetual right to free usage, modification, and redistribution. Github was chosen as a development and distribution platform to make it as easy as possible to obtain the software and participate in the development process by reporting bugs or contributing code[40]. Written in Python, the software does not depend on proprietary components. The choice of Jupyter notebooks as user interfaces facilitates the inspection of data at every analysis step and allows for tailoring and extending the pipeline specifically for the experimental system at hand. The **sdt-python** library[32] serves as the foundation and implements functionality to evaluate fluorescence microscopy data, such as single-molecule localization, diffusion analysis, fluorescence intensity analysis, color channel registration, colocalization analysis, and ROI handling.

In principle, single-particle tracking can be performed in one-, two- or three-dimensional systems. Here, the single-molecule analysis pipeline was tailored to the study of 2D mobile systems. This choice mirrors the availability of simple systems, such as planar-supported lipid bilayers (SLBs), to present mobile fluorescent probes. Such lipid bilayer systems are typically composed of two or more phospholipids moieties, where the bulk fraction determines the key physicochemical parameters of the SLB (such as phase and viscosity), and the minor fraction provides attachment sites for biomolecules. These attachment sites can be biotinylated phospholipids for avidin- or streptavidin-based protein platforms or nickel-NTA conjugated phospholipids for protein platforms with histidine tags[41]. The choice of the appropriate platform for linking proteins to the SLB depends on the scientific question. Readers can refer to the literature[16,38,42] for examples of successfully employed strategies. The density of probes in the sample should be sufficiently low to avoid overlapping point spread functions; typically, less than 0.1 molecules per µm² are recommended. See the representative results section (in particular, **Figure 6**) for an example showing a suitable probe density. The analysis method is also applicable to single fluorescently labeled protein molecules diffusing in the plasma membrane of live cells.

One critical aspect of smFRET experiments is the production and characterization of the FRET probes themselves. When choosing fluorophores for a FRET pair, their Förster radius should match the expected inter-dye distances[43]. Dyes resistant to photobleaching are preferred as they yield long time traces. However, for elevated bleaching rates, one fluorophore species can be utilized to recognize multiemitter events originating from colocalized molecules via stepwise photobleaching analysis; see step 8.1.4 in the protocol section. Fluorophore pairs should be site-specifically and covalently attached to the molecules of interest, forming intra- or intermolecular FRET pairs.

Combining smFRET with other readily available techniques can increase its spatial resolution beyond the diffraction limit (via STED[44]). The smFRET tracking algorithm presented here widens the approach's applicability to new experimental settings and model systems. This includes studies of (i) kinetic changes in the stoichiometry of mobile biomolecules, (ii) dynamic association of mobile biomolecules, (iii) the rate of enzymatic reactions of freely diffusing reactants, and (iv) the kinetics of conformational changes of mobile biomolecules. The first two examples require model systems that show intermolecular FRET, i.e., donor and acceptor are conjugated to separate biomolecular entities of interest. The latter examples may make use of biosensors carrying donor and acceptor within the same molecular entity (intramolecular FRET).

Intramolecular FRET-based sensors can provide insight into intrinsic conformational changes of biomolecules[1–4], conformational changes caused by endogenous or external force load (molecular force sensors[16]), or ion concentrations in the nano-environment such as calcium[45] and pH[46]. Depending on the model system and the preferred anchoring platform, such smFRET events can either be tracked in 2D or 3D: (i) planar tracking of smFRET events can be employed for the quantification of receptor–ligand interaction times within a plasma membrane, the association of membrane-anchored signal amplification cascades, and the stoichiometry changes of surface receptors; (ii) volume tracking of smFRET events can be used for any intra- or intermolecular FRET probes in living cells or in *in vitro* reconstituted systems.

The smFRET tracking method was developed mainly with intramolecular FRET probes in mind. These probes feature a fixed and well-known number of fluorescent labels, a fact that was exploited to reject data from agglomerated and incorrectly synthesized (e.g., incompletely labeled) molecules, as well as from probes where one of the fluorophores has been photobleached. However, by adjusting the filtering steps, the method can also be applied to intermolecular FRET probes. For instance, instead of accepting only molecules featuring a single donor and a single acceptor fluorophore, one could examine the spatial trajectories of donor and acceptor dyes and select, for example, for co-diffusing donor-acceptor trajectories.

As the 3D-DAOSTORM algorithm has support for determining a signal's position along the optical axis via the astigmatism due to a cylindrical lens in the emission beam path, 3D experiments could be easily integrated into the analysis pipeline. In this case, the acceptor signal upon acceptor excitation would serve to determine the stoichiometry and the axial position. The analysis software can also be employed to evaluate data from experiments featuring immobilized probes by utilizing its large degree of automation and filtering schemes. In fact, smFRET efficiency datasets from Holliday junctions immobilized on gel-phase bilayers[38] were analyzed using an early version of the software.

ACKNOWLEDGMENTS:

This work was supported by the Austrian Science Fund (FWF) projects P30214-N36, P32307-B, and by the Vienna Science and Technology Fund (WWTF) LS13-030.

DISCLOSURES:

The authors declare no conflicts of interest.

REFERENCES:

1. McKinney, S. A., Déclais, A. -C., Lilley, D. M. J., Ha, T. Structural dynamics of individual holliday junctions. *Nature Structural Biology*. **10** (2), 93–97 (2002).

2. Wang, S., Vafabakhsh, R., Borschel, W. F., Ha, T., Nichols, C. G. Structural dynamics of potassium-channel gating revealed by single-molecule FRET. *Nature Structural & Molecular Biology*. **23** (1), 31–36 (2015).

3. Hellenkamp, B., Wortmann, P., Kandzia, F., Zacharias, M., Hugel, T. Multidomain structure and correlated dynamics determined by self-consistent FRET networks. *Nature Methods*. **14** (2), 174–180 (2016).

4. Kilic, S. et al. Single-molecule FRET reveals multiscale chromatin dynamics modulated by HP1α. *Nature Communications*. **9** (1), 235 (2018).

5. Stryer, L. Fluorescence energy transfer as a spectroscopic ruler. *Annual Review of Biochemistry*. **47** (1), 819–846 (1978).

6. Wu, P. G., Brand, L. Resonance energy transfer: Methods and applications. *Analytical Biochemistry*. **218** (1), 1–13 (1994).

7. Qiao, Y., Luo, Y., Long, N., Xing, Y., Tu, J. Single-molecular förster resonance energy transfer measurement on structures and interactions of biomolecules. *Micromachines*. **12** (5), 492 (2021).

8. Malkusch, N., Dörfler, T., Nagy, J., Eilert, T., Michaelis, J. smFRET experiments of the RNA polymerase II transcription initiation complex. *Methods*. **120**, 115–124 (2017).

9. Lee, J. -B. et al. Single-molecule views of MutS on mismatched DNA. *DNA repair*. **20**, 82–93 (2014).

10. Phelps, C., Israels, B., Jose, D., Marsh, M. C., Hippel, P. H. von, Marcus, A. H. Using microsecond single-molecule FRET to determine the assembly pathways of T4 ssDNA binding protein onto model DNA replication forks. *Proceedings of the National Academy of Sciences of the United States of America*. **114** (18), E3612–E3621 (2017).

11. Deindl, S., Zhuang, X. Monitoring conformational dynamics with single-molecule fluorescence energy transfer: Applications in nucleosome remodeling. *Methods in Enzymology*. **513**, 59–86 (2012).

12. Crawford, D. J., Hoskins, A. A., Friedman, L. J., Gelles, J., Moore, M. J. Single-molecule colocalization FRET evidence that spliceosome activation precedes stable approach of 5' splice site and branch site. *Proceedings of the National Academy of Sciences of the United States of America*. **110** (17), 6783–6788 (2013).

13. Wang, Y., Xiao, M., Li, Y. Heterogeneity of single molecule FRET signals reveals multiple active ribosome subpopulations. *Proteins*. **82** (1), 1–9 (2014).

14. Mori, T., Vale, R. D., Tomishige, M. How kinesin waits between steps. *Nature*. **450** (7170), 750–754 (2007).

15. Huppa, J. B. et al. TCR–peptide–MHC interactions in situ show accelerated kinetics and increased affinity. *Nature*. **463** (7283), 963–967 (2010).

16. Göhring, J. et al. Temporal analysis of T-cell receptor-imposed forces via quantitative single molecule FRET measurements. *Nature Communications*. **12** (1), 2502 (2021).

17. Hellenkamp, B. et al. Precision and accuracy of single-molecule FRET measurements—a multi-laboratory benchmark study. *Nature Methods*. **15** (9), 669 (2018).

18. Kapanidis, A. N. et al. Fluorescence-aided molecule sorting: Analysis of structure and interactions by alternating-laser excitation of single molecules. *Proceedings of the National Academy of Sciences of the United States of America*. **101** (24), 8936–8941 (2004).

19. Sakon, J. J., Weninger, K. R. Detecting the conformation of individual proteins in live cells. *Nature Methods*. **7** (3), 203–205 (2010).

20. McCann, J. J., Choi, U. B., Zheng, L., Weninger, K., Bowen, M. E. Optimizing methods to recover absolute FRET efficiency from immobilized single molecules. *Biophysical Journal*. **99** (3), 961–970 (2010).

21. Lee, N. K. et al. Accurate FRET measurements within single diffusing biomolecules using alternating-laser excitation. *Biophysical Journal*. **88** (4), 2939–2953 (2005).

22. Asher, W. B*. et al.* Single-molecule FRET imaging of GPCR dimers in living cells. *Nature Methods*. **18** (4), 397–405 (2021).

23. Joo, C., Ha, T. Single-molecule FRET with total internal reflection microscopy. *Cold Spring Harbor Protocols*. **2012** (12), 1223–1237 (2012).

24. Joo, C., Ha, T. Objective-type total internal reflection microscopy (excitation) for single-molecule FRET. *Cold Spring Harbor Protocols*. **2012** (11), 1189–1191 (2012).

25. Joo, C., Ha, T. Objective-type total internal reflection microscopy (emission) for single-molecule FRET. *Cold Spring Harbor Protocols*. **2012** (11), 1192–1194 (2012).

26. Allan, D. B., Caswell, T., Wel, C. M. van der, Dimiduk, T. *Soft-matter/pims: PIMS v0.5*. https://github.com/soft-matter/pims (2020).

27. Anaconda, Inc. Miniconda. https://docs.conda.io/en/latest/miniconda.html (2021).

28. conda-forge community. The conda-forge project: community-based software distribution built on the conda package format and ecosystem. doi: 10.5281/zenodo.4774217 (2015).

29. JupyterLab Contributors Notebooks — JupyterLab documentation. https://jupyterlab.readthedocs.io/en/stable/user/notebook.html (2021).

30. Babcock, H., Sigal, Y. M., Zhuang, X. A high-density 3D localization algorithm for stochastic optical reconstruction microscopy. *Optical Nanoscopy*. **1**, 6 (2012).

31. Gao, Y., Kilfoil, M. L. Accurate detection and complete tracking of large populations of features in three dimensions. *Optics Express*. **17** (6), 4685 (2009).

32. Schrangl, L. sdt-python: Python library for fluorescence microscopy data analysis (v17.1). *Zenodo*. doi: 10.5281/zenodo.5509791 (2021).

33. Crocker, J. C., Grier, D. G. Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science*. **179** (1), 298–310 (1996).

34. Preibisch, S., Saalfeld, S., Schindelin, J., Tomancak, P. Software for bead-based registration of selective plane illumination microscopy data. *Nature Methods*. **7** (6), 418–419 (2010).

35. Bradski, G. The OpenCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer.* **25** (11), 120–123 (2000).

36. Allan, D. B., Caswell, T., Keim, N. C., van der Wel, C. M., Verweij, R. W. Soft-matter/trackpy: Trackpy v0.5.0. *Zenodo*. doi: 10.5281/zenodo. 4682814 (2021).

37. Killick, R., Fearnhead, P., Eckley, I. A. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*. **107** (500), 1590–1598 (2012).

38. Schrangl, L., Göhring, J., Schütz, G. J. Kinetic analysis of single molecule FRET transitions without trajectories. *The Journal of Chemical Physics*. **148** (12), 123328 (2018).

39. Santoso, Y., Torella, J. P., Kapanidis, A. N. Characterizing single-molecule FRET dynamics with probability distribution analysis. *ChemPhysChem*. **11** (10), 2209–2219 (2010).

40. Schrangl, L. Single-molecule FRET analysis software (3.0). *Zenodo*. doi: 10.5281/zenodo.5115967 (2021).

41. Nye, J. A., Groves, J. T. Kinetic control of histidine-tagged protein surface density on supported lipid bilayers. *Langmuir*. **24** (8), 4145–4149 (2008).

42. Platzer, R. et al. Unscrambling fluorophore blinking for comprehensive cluster detection via photoactivated localization microscopy. *Nature Communications*. **11** (1), 4993 (2020).

43. The molecular probes handbook: A guide to fluorescent probes and labeling technologies. Johnson, I., Spence, M. (Eds), Life Technologies Corporation (2010).

44. Szalai, A. M. et al. Super-resolution imaging of energy transfer by intensity-based STED-FRET. *Nano Letters*. **21** (5), 2296–2303 (2021).

45. Miyawaki, A. et al. Fluorescent indicators for Ca^{2+} based on green fluorescent proteins and calmodulin. *Nature*. **388** (6645), 882–887 (1997).

46. Zhai, B., Zhai, S., Hao, R., Xu, J., Liu, Z. A FRET-based two-photon probe for in vivo tracking of pH during a traumatic brain injury process. *New Journal of Chemistry*. **43** (43), 17018–17022 (2019).

Figure 1

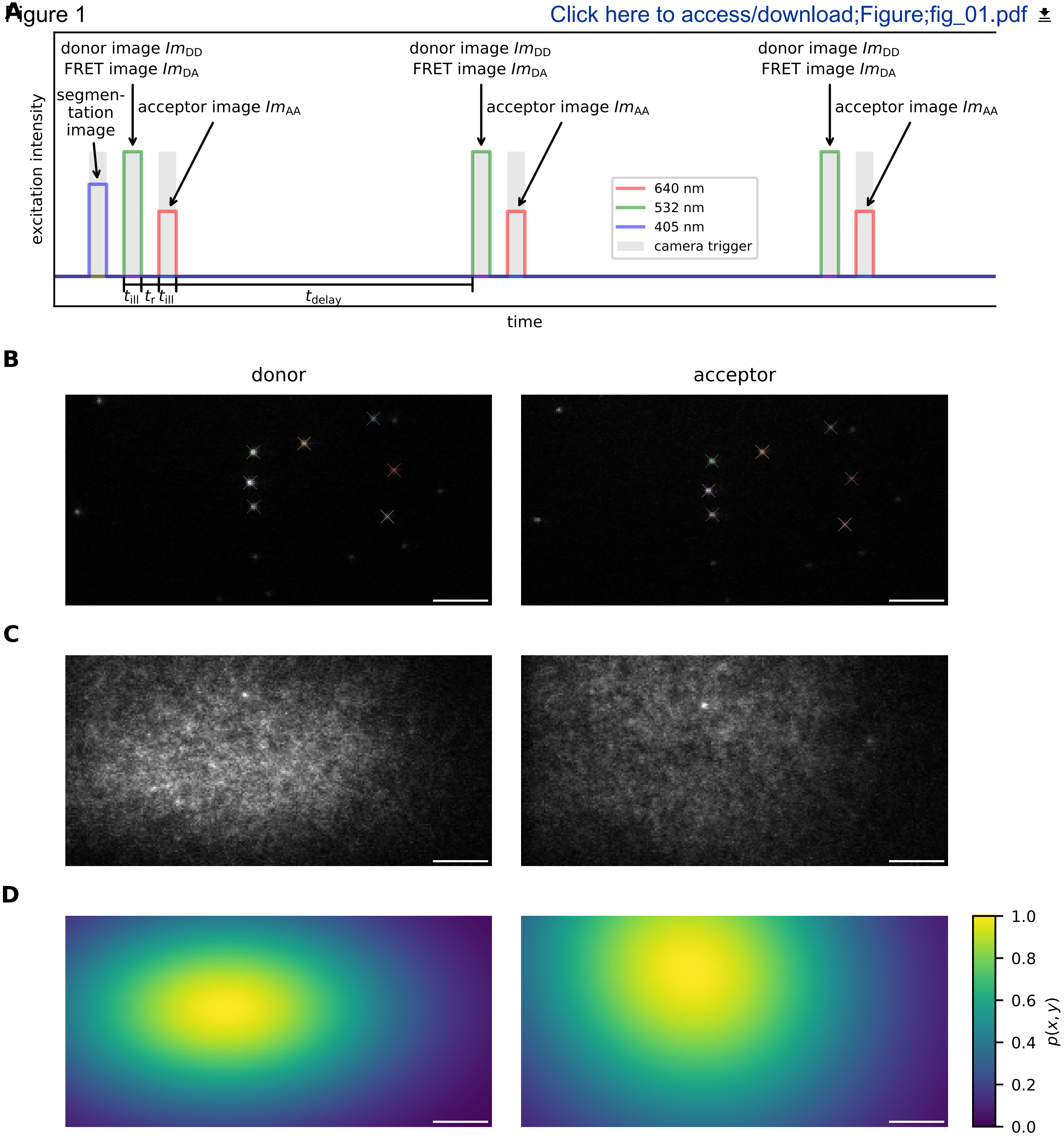

A
donor image Im_{DD}
FRET image Im_{DA}
segmen-
tation
image
acceptor image Im_{AA}
donor image Im_{DD}
FRET image Im_{DA}
acceptor image Im_{AA}
donor image Im_{DD}
FRET image Im_{DA}
acceptor image Im_{AA}
640 nm
532 nm
405 nm
camera trigger
excitation intensity
t_{ill}
t_r
t_{ill}
t_{delay}
time
B
donor
acceptor
C
D
1.0
0.8
0.6
0.4
0.2
0.0
$p(x, y)$

Figure 2

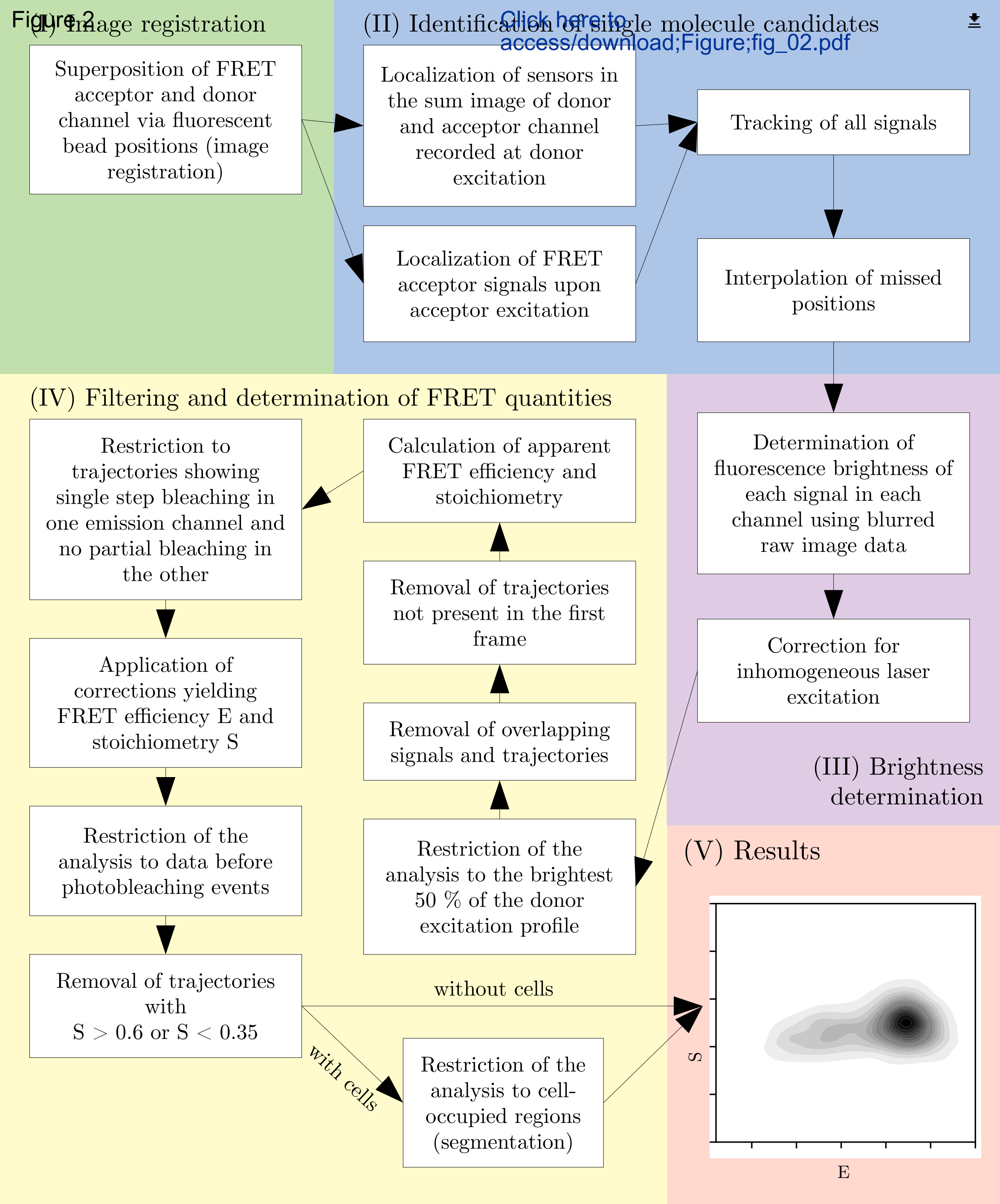

(I) Image registration
Superposition of FRET acceptor and donor channel via fluorescent bead positions (image registration)
(II) Identification of single molecule candidates
Localization of sensors in the sum image of donor and acceptor channel recorded at donor excitation
Localization of FRET acceptor signals upon acceptor excitation
Tracking of all signals
Interpolation of missed positions
(III) Brightness determination
Determination of fluorescence brightness of each signal in each channel using blurred raw image data
Correction for inhomogeneous laser excitation
(IV) Filtering and determination of FRET quantities
Restriction of the analysis to the brightest 50 % of the donor excitation profile
Removal of overlapping signals and trajectories
Removal of trajectories not present in the first frame
Calculation of apparent FRET efficiency and stoichiometry
Restriction to trajectories showing single step bleaching in one emission channel and no partial bleaching in the other
Application of corrections yielding FRET efficiency E and stoichiometry S
Restriction of the analysis to data before photobleaching events
Removal of trajectories with S > 0.6 or S < 0.35
without cells
with cells
Restriction of the analysis to cell-occupied regions (segmentation)
(V) Results
S
E

Figure 3

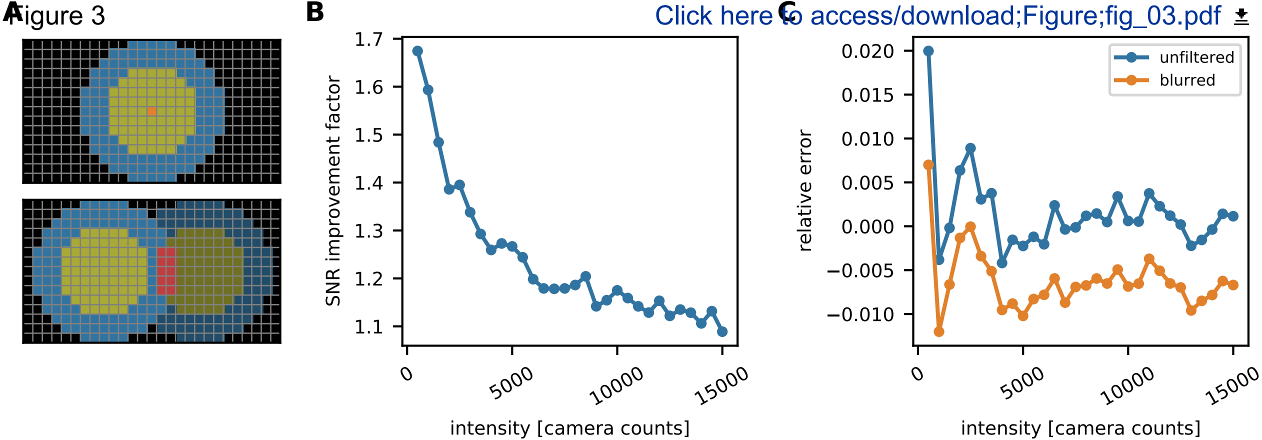

A
B
C
1.7
1.6
1.5
1.4
1.3
1.2
1.1
SNR improvement factor
0
5000
10000
15000
intensity [camera counts]
0.020
0.015
0.010
0.005
0.000
−0.005
−0.010
relative error
unfiltered
blurred
0
5000
10000
15000
intensity [camera counts]

Figure 4

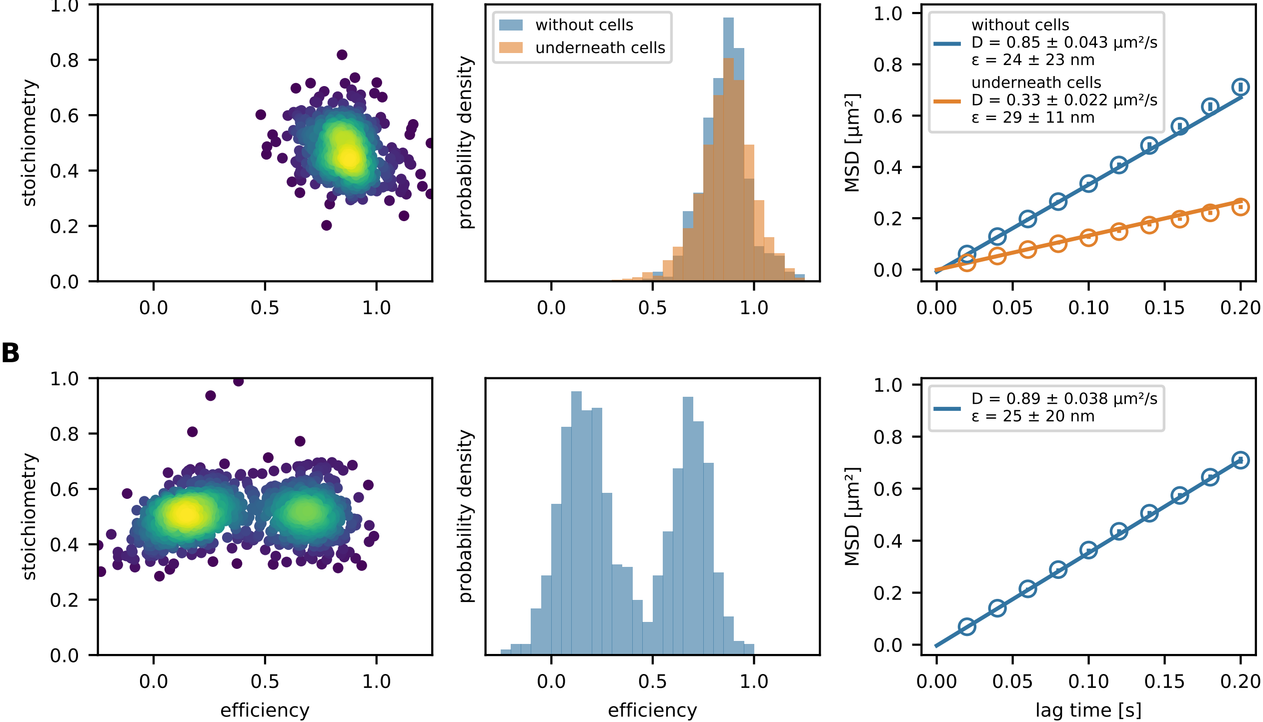

A
stoichiometry
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
without cells
underneath cells
probability density
0.0
0.5
1.0
without cells
D = 0.85 ± 0.043 μm²/s
ε = 24 ± 23 nm
underneath cells
D = 0.33 ± 0.022 μm²/s
ε = 29 ± 11 nm
MSD [μm²]
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.05
0.10
0.15
0.20
B
stoichiometry
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
efficiency
probability density
0.0
0.5
1.0
efficiency
D = 0.89 ± 0.038 μm²/s
ε = 25 ± 20 nm
MSD [μm²]
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.05
0.10
0.15
0.20
lag time [s]

Figure 5

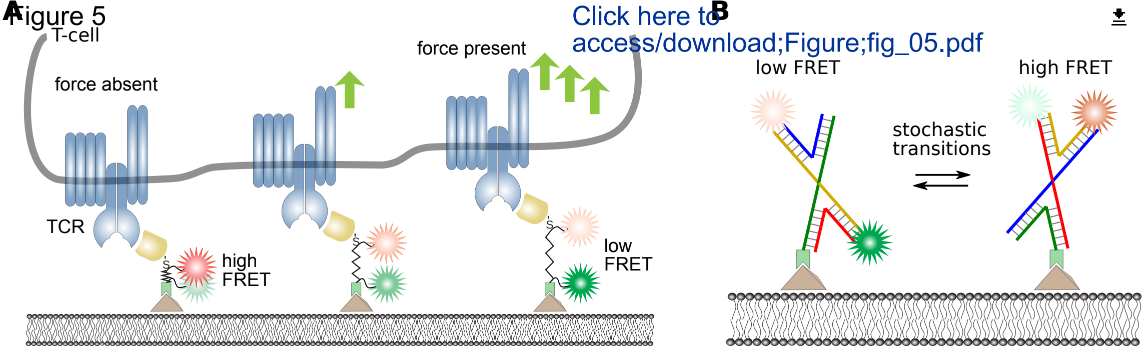
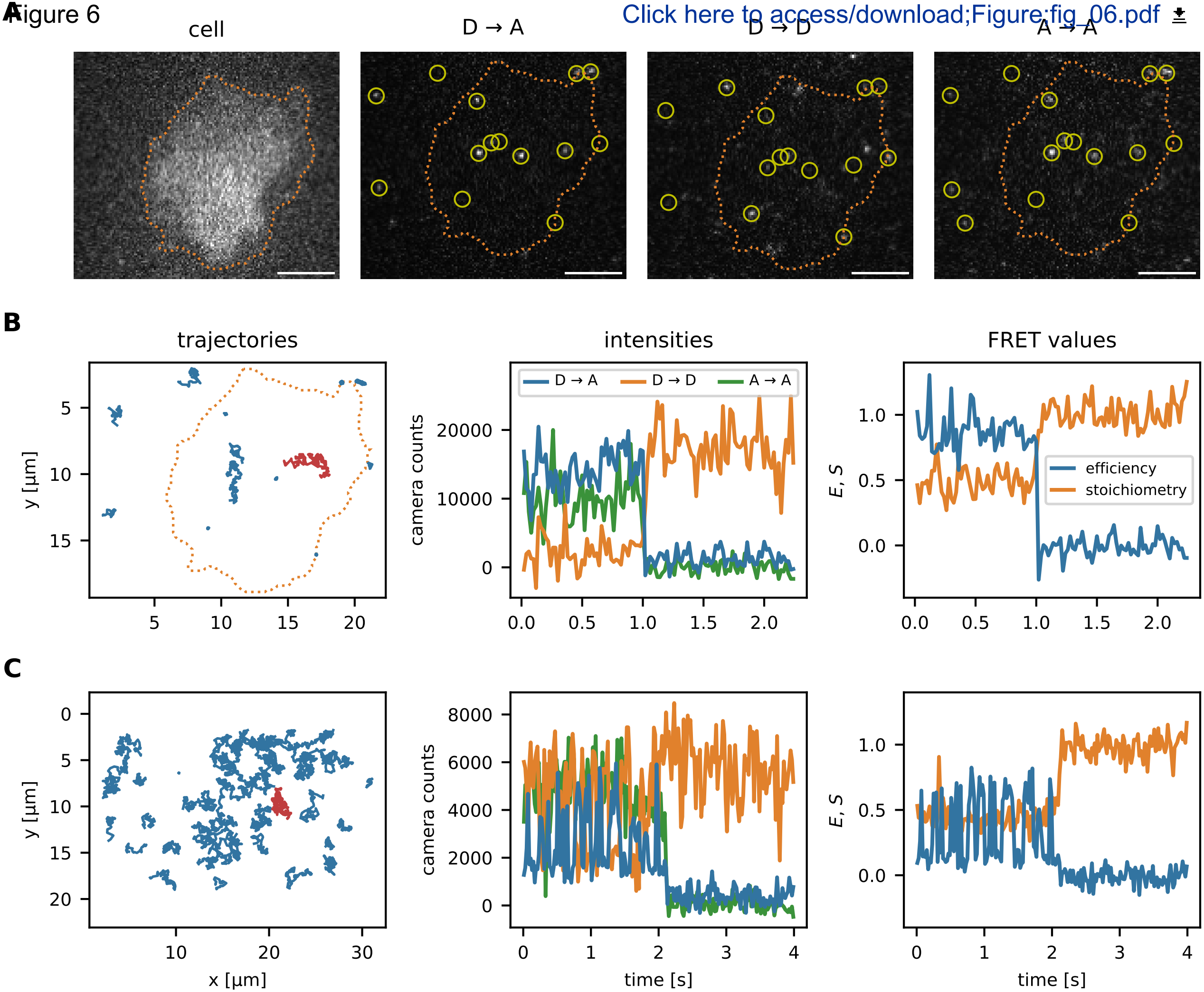

A
T-cell
force absent
TCR
S
high
FRET
S
force present
S
low
FRET
B
low FRET
high FRET
stochastic
transitions

Figure 6


A
cell
D → A
D → D
A → A
B
trajectories
intensities
FRET values
y [µm]
5
10
15
5
10
15
20
camera counts
20000
10000
0
D → A
D → D
A → A
0.0
0.5
1.0
1.5
2.0
E, S
1.0
0.5
0.0
efficiency
stoichiometry
0.0
0.5
1.0
1.5
2.0
C
y [µm]
0
5
10
15
20
10
20
30
x [µm]
camera counts
8000
6000
4000
2000
0
0
1
2
3
4
time [s]
E, S
1.0
0.5
0.0
0
1
2
3
4
time [s]

Dear Dr. Iyer

We have addressed all requested changes in the manuscript

Best wishes

Gerhard Schütz

1 Localization and tracking of single molecules (01. Tracking.ipynb)

1.1 Import and setup of required Python packages

[1]:

```
%matplotlib widget

import matplotlib.pyplot as plt
plt.ioff()

from smfret_analysis import Tracker, print_info
```

Press Shift+Enter to execute the cell.

1.2 Description of datasets

[2]:

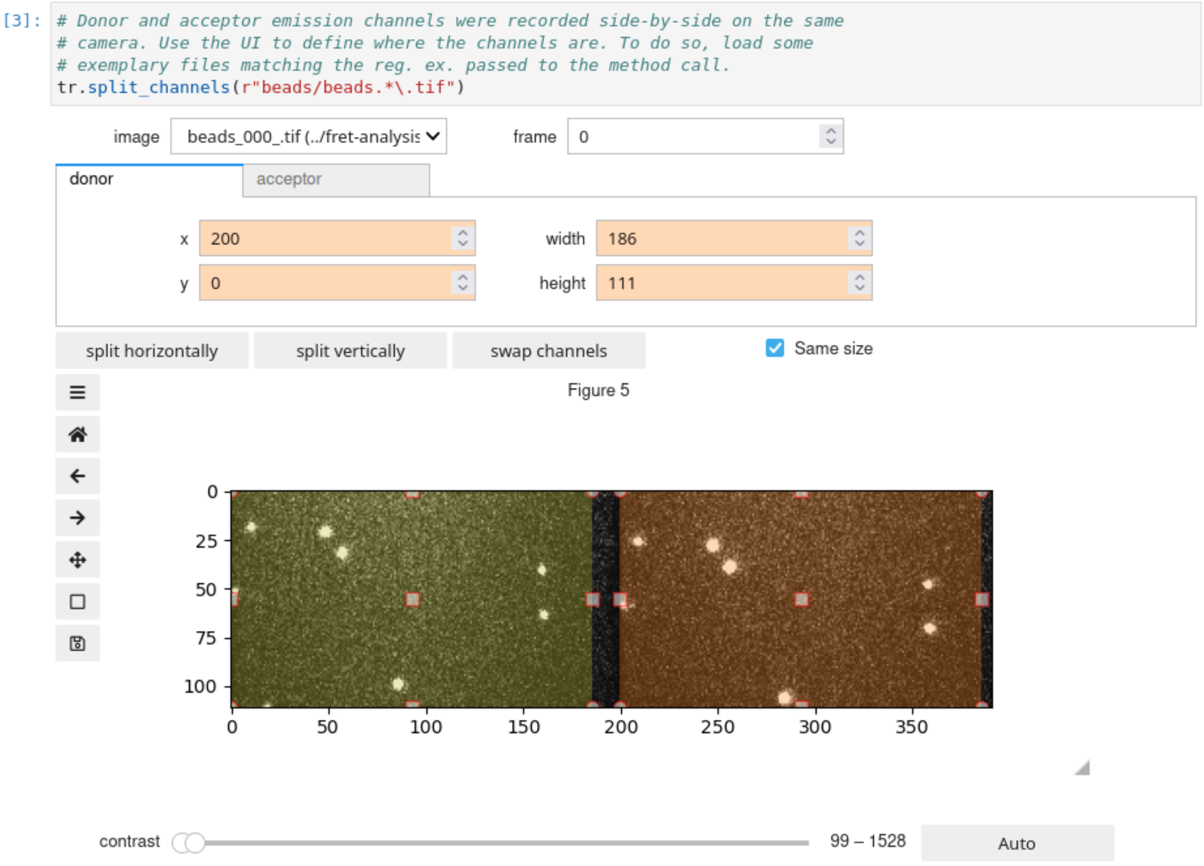
```
# Create a Tracker instance

# If starting fresh, set load = False. If there is already saved data from a previous run,
# load = True can be used to load it.
load = False

if not load:
    # Directory containing raw data
    data_dir = "../fret-analysis/sm-data"

    # Specify
    # - excitation sequence, where "c" means cell contours, "a" means acceptor excitation,
    #   and "d" means donor excitation
    # - data directory
    tr = Tracker("se + da * 300 + s", data_dir=data_dir)

    # Add datasets by giving
    # - an identifier
    # - a regular expression describing the file names relative to data_dir.
    #   Use forward slashes as path separators even on Windows.
    tr.add_dataset("DPPC control cells",
                   r"^DPPC_ctrl-J4/cells.*\.tif")
    tr.add_dataset("DPPC control no-cells",
                   r"^DPPC_ctrl-J4/no-cells.*\.tif")
    # If donor and acceptor emission channels are recorded by separate cameras, resulting
    # in separate video files, the second argument is a regular expression for the donor,
    # the third argument a regular expression for the acceptor emission. First donor file
    # will be matched with first acceptor file, second with second, etc., so make sure
    # they have the same sorting order.
    # tr.add_dataset("DPPC control cells",
    #                r"^DPPC_ctrl-J4/cells.*_donor\.tif",
    #                r"^DPPC_ctrl-J4/cells.*_acceptor\.tif")
    # tr.add_dataset("DPPC control no-cells",
    #                r"^DPPC_ctrl-J4/no-cells.*_donor\.tif",
    #                r"^DPPC_ctrl-J4/no-cells.*_acceptor\.tif")

    # Special datasets for registration etc. If donor and acceptor emission are recorded
    # in separate files, again use two regular expressions.
    tr.add_special_dataset("registration", r"beads/beads_.*\.tif")
    tr.add_special_dataset("donor-profile", "profile/green_.*\.tif")
    tr.add_special_dataset("acceptor-profile", "profile/red_.*\.tif")
    # tr.add_special_dataset("donor-only", r"^DO/no-cells.*\.tif")
    # tr.add_special_dataset("acceptor-only", r"^AO/no-cells.*\.tif")
else:
    # Load
    tr = Tracker.load()
```

1. Setting `load = True` will load data previously saved using the `save()` method in the at the end of the notebook. Execute the cell (`Shift+Enter`) and go to any cell below to redo the analysis from there. For the initial analysis, use `load = False`.
2. Point `data_dir` to the directory containing your raw data. When using backslashes on Windows to separate folders, prefix the string with `r`, e.g. `r"C:\path\to\data"` to specify that the backslashes are to be taken literally.
3. Create a `Tracker` instance named `tr`. The first argument to the constructor call is a string describing the excitation sequence. Each letter corresponds to one frame. Use `d` for donor excitation, `a` for acceptor excitation, and `s` for segmentation image. Any other letters can also be used for costum frames, but are ignored by the analysis. `+` and `*` operators can be used to concatenate and repeat sequence parts. For instance, `"s + da * 300 + s"` describes a sequence of one segmentation

image excitation followed by 300 repeats of donor and acceptor alternating excitation, and another segmentation image excitation in the end. The sequence is automatically repeated as necessary, so `"da"` is equivalent to `"dadadadada"`.

4. Add datasets using `tr.add_dataset(…)` (multiple times if needed). The first argument is a string identifying the dataset. Choosing descriptive denomination is recommended, e.g. `"condition1 no-cells"`. The second (and potentially third) argument is a regular expression pattern (see, for instance, the documentation of Python's standard library's `re` module)[1] matching the names of raw data files which constitute the respective datasets. In regular expressions, a fullstop `.` serves as a wildcard for any character, and an asterisk `*` matches zero or more of the preceding character. Consequently, `.*` translates to "any number of any character." A literal fullstop is matched by `\.`. The patterns should be relative to `data_dir` and using forward slashes `/` to separate folders, even on Windows operating systems. For instance, to match all files which start with `no-cells` and have extension `tif` in a subfolder named `condition1`, use `"condition1/no-cells.*tif"`. In case donor and acceptor emission were recorded to the same files, the pattern should match these files. If they were written to separate files, the second argument describes the donor emission, and the third describes the acceptor emission.
5. Similarly, add datasets used for calculating correction coefficients using `tr.add_special_dataset(…)`. Here, the dataset identifiers can be either `"registration"`, `"donor-profile"`, `"acceptor-profile"`, `"donor-only"`, `"acceptor-only"`, or `"multi-state"` for fiducial markers, donor or acceptor emission using the densly labeled sample (can be the same files if the acceptor is photobleached), donor-only single-molecule sample for leakage correction, acceptor-only sample for cross-excitation correction, or a multi-state sample for excitation and detection efficiency correction.
6. Execute the cell.

1.3 Crop channels



[3]:
```
# Donor and acceptor emission channels were recorded side-by-side on the same
# camera. Use the UI to define where the channels are. To do so, load some
# exemplary files matching the reg. ex. passed to the method call.
tr.split_channels(r"beads/beads.*\.tif")
```

image: beads_000_.tif (../fret-analysis

frame: 0

donor | acceptor

x: 200

width: 186

y: 0

height: 111

☑ Same size

contrast: 99 – 1528

Select the emission channels' regions in the raw images. Only applicable if both donor and acceptor emission were recorded side by side, skip otherwise.

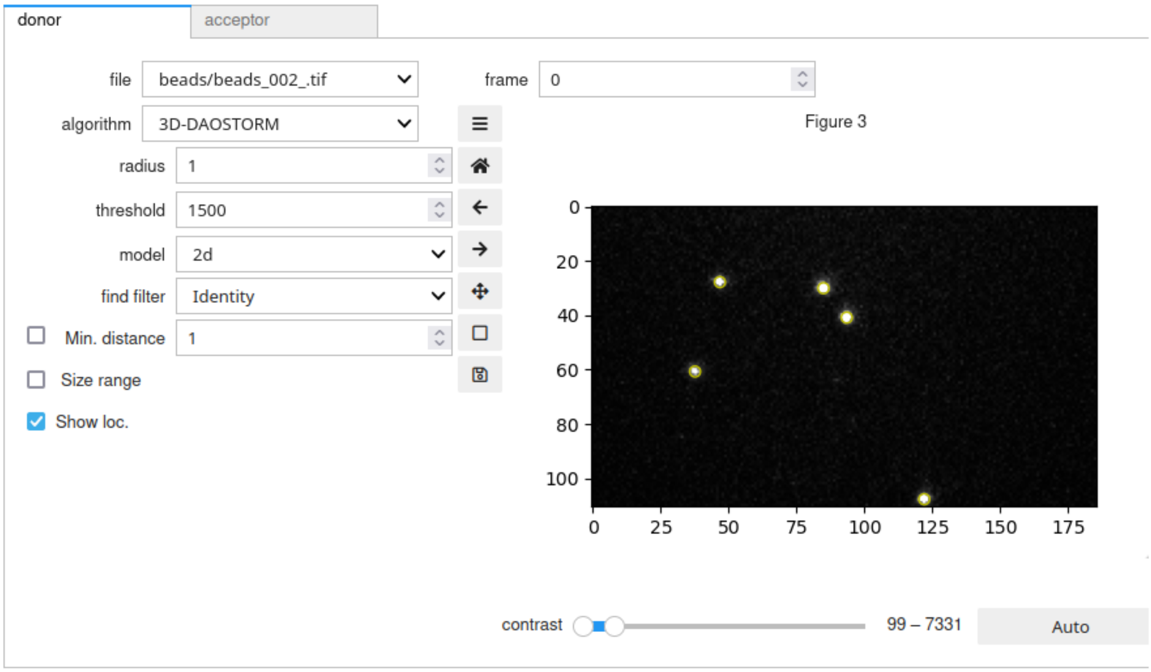
1. The argument to `tr.split_channels(…)` should be a regular expression matching representative files. For instance, to use the images of the fiducial markers, assuming they are TIFF files located in

a subfolder name `beads`, use `"beads/.*tif"`

2. Execute the cell, which will bring up UI elements to select the channels' regions in the image files.
3. Depending on the layout, press `split horizontally` or `split vertically` to split in the middle. The donor channel is marked by an orange rectangle, the acceptor channel by a yellow one. If you want to swap the channels, press the `swap channels` button
4. Fine-tuning can be achieved by manually entering the coordinates in the appropriate boxes or by dragging the rectangles. Note that you can only change the rectangle of channel that is currently selected via the tab bar on the top. For instance, to resize the acceptor rectangle, the acceptor tab needs to be activate by pressing the respective button.

1.4 Set parameters for localization of fiducial markers for image registration

[4]:
```
# Set localization options for fiducial markers, which are used below to
# calculate the coordinate transform between donor and acceptor channels.
tr.set_registration_loc_opts()
```



donor | acceptor

file: beads/beads_002_.tif

frame: 0

algorithm: 3D-DAOSTORM

radius: 1

threshold: 1500

model: 2d

find filter: Identity

☐ Min. distance: 1

☐ Size range

☑ Show loc.

contrast: 99 – 7331

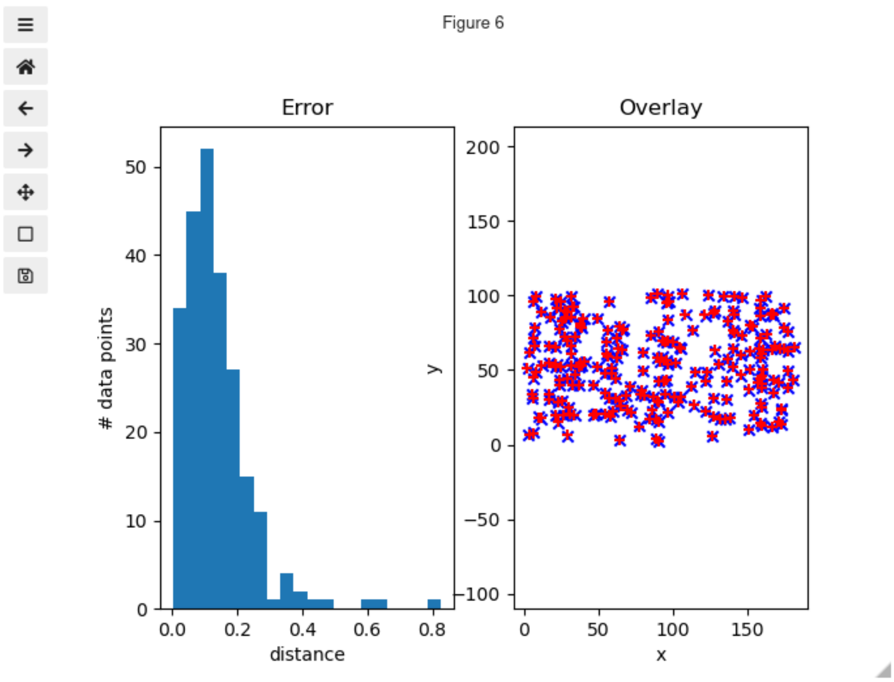
Auto

1. Execute `tr.set_registration_loc_opts()`.
2. A widget will appear that allows for establishing the localization algorithm and parameters.
3. Browse images and adjust until satisfied with the outcome. Do so for both donor and acceptor channels.

1.5 Image registration

[5]:

```
# Calculate the coordinate transform from bead localizations
tr.calc_registration()
```

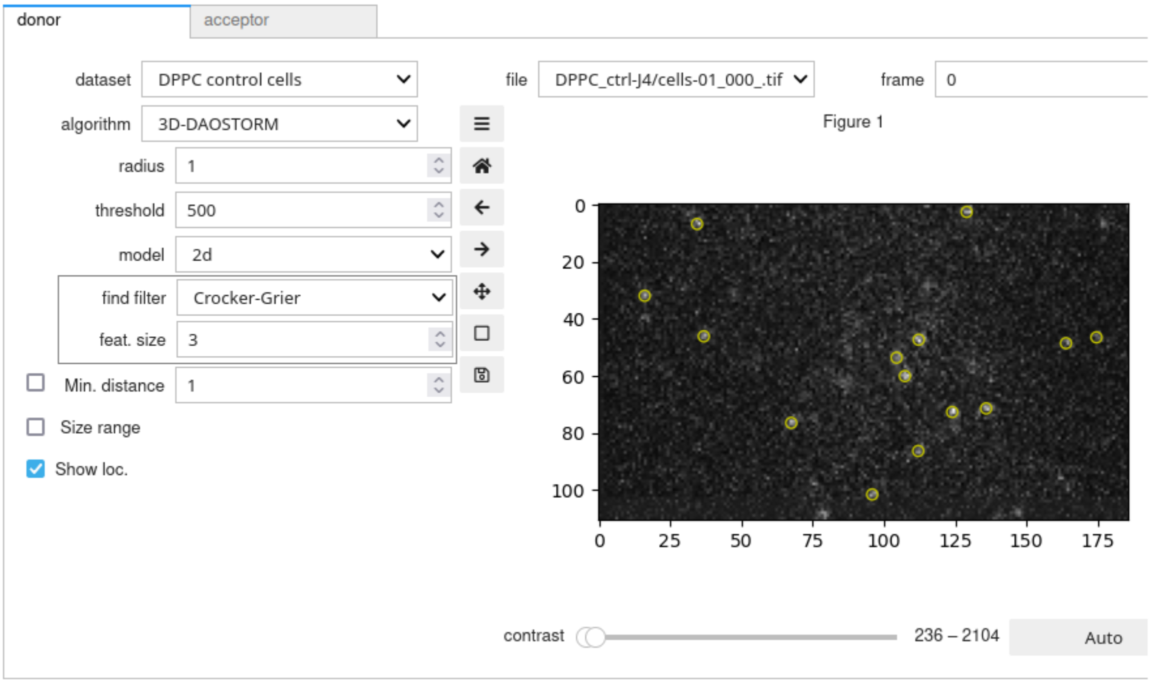


Executing `tr.calc_registration()` will first localize all fiducial markers, then calculate the transformation mapping the donor channel coordinates onto the acceptor channel coordinates and vice versa. When finished, a plot is will show the result. In case this is not satisfactory, the algorithm can be tuned via optional parameters. Please refer to the documentation supplied with the code for details.

1.6 Set parameters for localization of FRET probes

[6]:

```
tr.set_loc_opts()
```



dataset: DPPC control cells

file: DPPC_ctrl-J4/cells-01_000_.tif

frame: 0

algorithm: 3D-DAOSTORM

radius: 1

threshold: 500

model: 2d

find filter: Crocker-Grier

feat. size: 3

□ Min. distance: 1

□ Size range

☑ Show loc.

contrast: 236 – 2104

1. Execute `tr.set_loc_opts()` to display a user interface that permits setting the localization algorithm parameters for detecting the FRET probes.
2. The `donor` tab displays the sum of donor and acceptor emission upon donor excitation. This allows for localizing the probes irrespectively of the FRET efficiency, since the total number of emitted photons remains approximately constant.
3. The `acceptor` tab shows the acceptor emission upon acceptor excitation.
4. Set the localization parameters for each channel individually.

1.7 Localize FRET probes

```
[*]: # Run localization algorithm. This can take a while.
     tr.locate()
```

```
Locating DPPC_ctrl-J4/cells-01_012_.tif (2/25)
```

Execute `tr.locate()` to localize all FRET signals. This can take some time, depending on the data quality and available computing power.

1.8 Tracking

```
[*]: # Track the localizations.
     tr.track(feat_radius=4, bg_frame=3, link_radius=5, link_mem=3, min_length=4, bg_estimator="mean")
```

```
Tracking DPPC_ctrl-J4/cells-01_032_.tif (5/25)
```

The following parameters for the tracking and fluorescence intensity measurement algorithms have to be set:

1. `feat_radius`: Radius (in pixels) of a disk large enough to contain the whole point spread function. This is used for measuring the fluorescence intensity—all pixels' intensities within the disk are added up and corrected for background—as well as for detecting whether point spread functions overlap, i.e. their distance is less than `2 * feat_radius + 1`, in which case the signal can be filtered out later.
2. `bg_frame`: A ring of `bg_frame` pixels around the disk described by `feat_radius` is used for local background determination. Pixels which are part of another probe's point spread function are automatically excluded.
3. `link_radius`: Maximally allowed distance for probe movement between consecutive frames in pixels. If chosen too low, localizations may not be linked properly. If too large, the algorithm can fail due to high complexity.
4. `link_mem`: Maximum gap size within a trajectory, i.e., number of consecutive frames where a signal can be absent. Set at least to 1 to permit tracks to continue after one constituent of the FRET pair has been bleached.
5. `bg_estimator`: A statistic to determine the local background from the pixel values selected via the `bg_frame`. `"mean"` is recommended, `"median"` is also an option.
6. For further options, consult the documentation supplied with the code.

Execute the cell to perform tracking and to determine fluorescence intensities. This can take some time, depending on the data quality and available computing power.

1.9 Extraction of auxiliary image data

```
[9]: # Get the images of the cells
     tr.extract_segment_images()

     # Get the laser profiles from additional recordings of bulk samples
     tr.make_flatfield("donor", bg=200, smooth_sigma=10)
     tr.make_flatfield("acceptor", bg=200, smooth_sigma=10)
```

1. `tr.extract_segment_images()` extracts images for segmentation (marked by the letter `s` in the excitation sequence) from image sequences.

2. `tr.make_flatfield(…)` determines the excitation light profiles across the field of view. Call once with `"donor"` as the first argument and once with `"acceptor"` as the first argument to do the computation for both channels. To compute the profile, the frame number selected via the `frame` argument is loaded from each image in the `donor-profile` or `acceptor-profile` special dataset. Subsequently, the pixel-by-pixel mean of these images is calculated, and the camera baseline given by the `bg` parameter is subtracted. The result is smoothed using a Gaussian blur with a σ of `smooth_sigma` pixels to smooth out sample impurities.
3. Execute the cell.

1.10 Save data

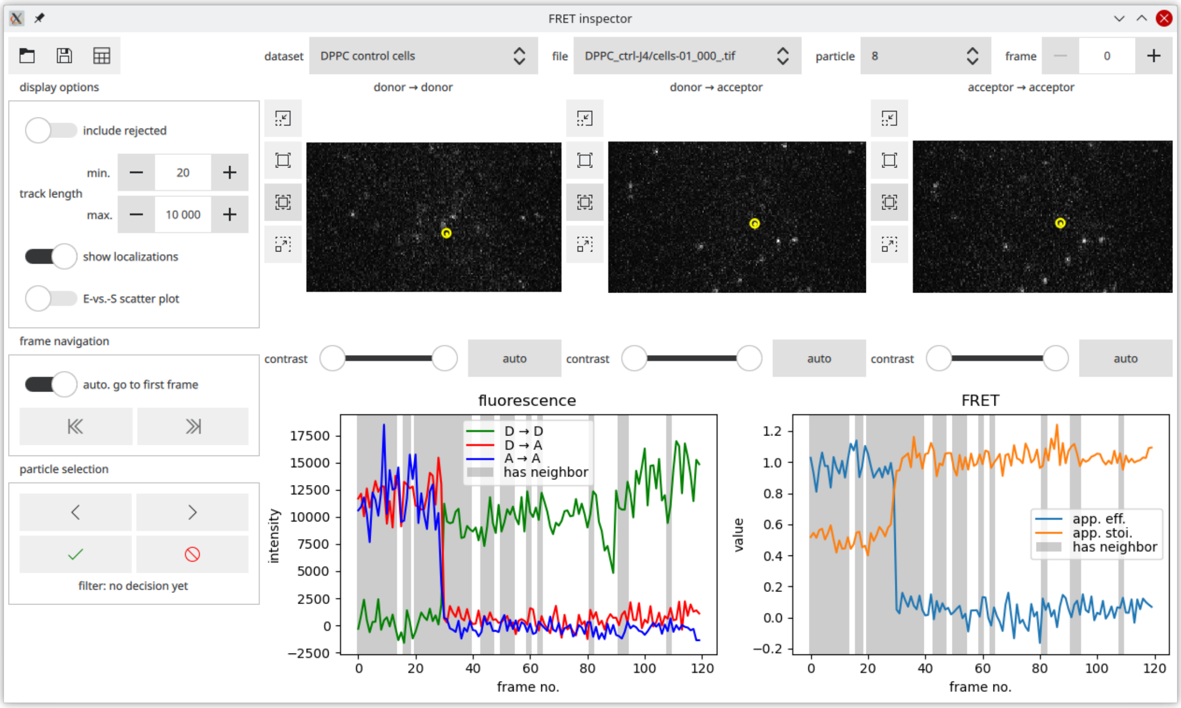
[10]:
```
# Save data
tr.save()
```

Executing `tr.save()` to save data to disk. This will create several files prefixed with `tracking`. To change the prefix, e.g. if there are multiple datasets in the same folder, pass a corresponding string as a parameter to `tr.save()`. Don't forget to pass the same prefix to `tr = Tracker.load()` in cell number 2 if rerunning parts of the localization and tracking process. The generated `.yaml` file can be openend in a text editor to inspect the parameters which were used. The `.h5` files contain single-molecule localization and tracking data. Contents may be loaded using the *pandas* Python library.

1.11 Inspection of data (optional)

Results can be inspected and modified at any point of the analysis. The tracker instance `tr` holds all data. Single-molecule tracking results can be obtained via the `tr.track_data` attribute. See the code documentation for information on the data structure.

2 Visualization of FRET trajectories (optional)



FRET inspector

dataset: DPPC control cells

file: DPPC_ctrl-J4/cells-01_000_.tif

particle: 8

frame: 0

track length min.: 20

track length max.: 10 000

filter: no decision yet

1. In an Anaconda prompt, execute

```
python -m smfret_analysis.inspector_gui
```

in the root directory. A new window will open, which may take a few seconds.

2. Press the `open` button (top left) button and select the `.yaml` file which was created using the tracking Jupyter notebook. Data will be loaded, which can take some seconds.

3. Top row dropdown menus permit selecting the track and frame number to display. Note that numbers start at 0. Frame numbers only count donor excitation frames. The nearest acceptor excitation frame is choosen automatically.

4. `display options` allow for choosing whether to display rejected tracks (see below), rejecting tracks based on their length, enabling and disabling the display of localizations and tracks on top of the raw images, and choosing whether to draw efficiency-vs.-stoichiometry scatter plots or time traces in the bottom right diagram.

5. Under `frame navigation` it is possible to toggle whether or not to go automatically to the first frame of a track after selecting it. The arrow buttons allow for navigating to the first and last frame of the currently selected track.

6. Using the `particle selection` buttons it is possible to proceed to the next or previous track and mark the current track either as manually selected or rejected.

7. If manual filtering was performed, use the `save` button (top left, second button) to save the selection.

8. Data can be exported to a spread sheet using the third button in the top-left corner. Note that this can takes several tens of seconds. Meanwhile, the application appears to be frozen.

3 Analysis and filtering of single-molecule data (`02. Analysis.ipynb`)

3.1 Import and setup of required Python packages

```
[1]: %matplotlib widget

     import matplotlib.pyplot as plt
     plt.ioff()
     import numpy as np

     from smfret_analysis import Analyzer, DensityPlots, print_info
```

Press `Shift+Enter` to execute the cell.

3.2 Print information about used software versions

```
[2]: print_info()
```

```
smFRET analysis software version 2.1
(git revision 2.2-11-gaa76315)
Output version 13
Using sdt-python version 16.1
```

Execute cell to print version of the FRET analysis software and underlying libraries.

3.3 Load data

```
[3]: # Create an Analyzer instance. This will load the tracking data.
     ana = Analyzer()
```

Execute cell to create an `Analyzer` instance named `ana`. This loads the tracking data from save files created by the tracking notebook. The `Analyzer()` call take can one optional argument: the file name prefix. Pass whatever was passed to the `save` function in the tracking notebook.

3.4 Choose datasets to visualize after each filtering step (optional)

```
[4]: # Show apparent E vs. S plots after filtering steps. Here we use the JupyterLab
     # Sidecar class (https://github.com/jupyter-widgets/jupyterlab-sidecar) for
     # display, but it would also possible to output to a normal notebook cell.

     # Which datasets to plot.
     plot_keys = []  # Use empty list to disable plotting.
     # plot_keys = [k for k in ana.analyzers if k.endswith(" cells")]  # Plot datasets with cells

     if plot_keys:
         from sidecar import Sidecar
         sc = Sidecar(title="Filtered plots")

         dp = DensityPlots(ana, plot_keys)
         with sc:
             display(dp)
```

Add the datasets' names (as specified by the first argument to `Tracker.add_dataset` calls in the tracking notebook) to the `plot_keys` list, e.g. `plot_keys = ["condition1 cells", "condition2 cells"]`. Leave the list empty the disable plots, i.e., `plot_keys = []`. Execute cell to pop up a new panel in the browser window which will be used for plotting. Note that plotting only works if the `sidecar` Python package is installed and set up (see main text, protocol step 1.7).

3.5 Visualize unfiltered datasets (optional)

```
[5]: # Plot initial data

     # Due to typically large amounts of data, this may take a few tens of seconds
     if plot_keys:
         ana.calc_fret_values(a_mass_interp="next")
         dp.dscatter("Initial")  # Scatter plot
         # dp.contourf("Initial")  # Contour plot
         # dp.colormesh("Initial")  # Continous color
```

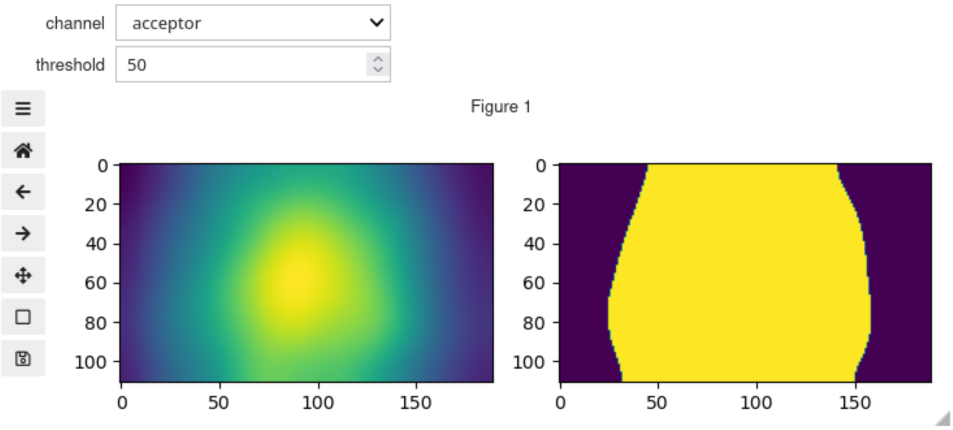
Execute cell to display apparent FRET efficiency vs. stoichiometry plots for the unfiltered datasets selected above. Plots are drawn in the side panel created in the previous step. The plot type depends on which function call is used:

- `scatter(…)` produces a scatter plot of all signals. Points are color-coded according to density in E–S space.
- `contour(…)` creates a contour plot, depeciting discrete levels of point densities in E–S space.
- `colormesh(…)` displays continous color map of densities in E–S space.

For large sets this may take several seconds. If no datasets were selected for plotting, this does nothing. All plotting functions take the plot title as an argument.

3.6 Select region of adaquate excitation intensity (optional)

```
[6]: # Display laser beam profile and find threshold of brightest xx %
     # to ensure good SNR
     ana.find_beam_shape_thresh()
```



channel: acceptor

threshold: 50

After execution of `ana.find_beam_shape_thresh()`, a widget will display the donor or acceptor excitation light profile on the left panel. The right panel shows the regions with intensities above and below `threshold` (in percent of the maximum intensity), where `threshold` can be set using the respective input field. This serves to visualize the effect of the next filtering step.

3.7 Accept only adequately excited FRET probes (optional)

```
[7]: # Copy values from above to remove poorly illuminated datapoints
     # Manually copy the parameters from above here.
     ana.filter_beam_shape_region("donor", 50)

     if plot_keys:
         dp.dscatter("Beam shape")
```

`ana.filter_beam_shape_region(…)` accepts, upon execution, only signals that are within the region of sufficient excitation intensity to ensure an appropriate signal-to-noise ratio. The first argument specifies the channel (`"donor"` or `"acceptor"`), the second argument the minimum intensity as a percentage of the peak intensity. We typically filter based on donor excitation intensity and set a threshold of 45% to 50%. If `plot_keys` was set in step 3.4, the filtered data will be plotted.

3.8 Initial sanity checks (optional)

```
[8]: # Allow only tracks with with a high number (here, >75%) of datapoints where
     # PSFs don't overlap
     ana.query_particles("fret_has_neighbor == 0", min_rel=0.75)
     # For the remaining tracks, allow only those datapoints with non-overlapping
     # PSFs
     ana.query("fret_has_neighbor == 0")
     # Accept only tracks that are already visible in the beginning to record
     # full bleaching profile
     ana.present_at_start()

     if plot_keys:
         dp.dscatter("At start")
```

Execute the cell to perform the following:

- `ana.query_particles("fret_has_neighbor == 0", min_rel=0.75)` rejects all tracks that have a near neighbor in more that 25% of their datapoints, as these tracks would contain too many missing points.
- The call to `ana.query("fret_has_neighbor == 0")` removes datapoints with overlapping point spread functions and thus erroneous values from the remaining tracks. This essentially creates gaps in the trajectories, which are, however, gracefully handled in ensuing analysis steps.

- With `ana.present_at_start()`, only tracks which are already present in the first donor excitation frame are accepted to ensure that the full photobleaching sequence is captured. Note that this is only applicable if smFRET complexes are already present when starting the imaging process.

If `plot_keys` was set in step 3.4, the filtered data will be plotted.

3.9 Flatfield correction

```
[9]: # Apply flatfield correction in both channels
     ana.flatfield_correction()
```

Correct for inhomogeneous excitation intensities by executing `ana.flatfield_correction()`.

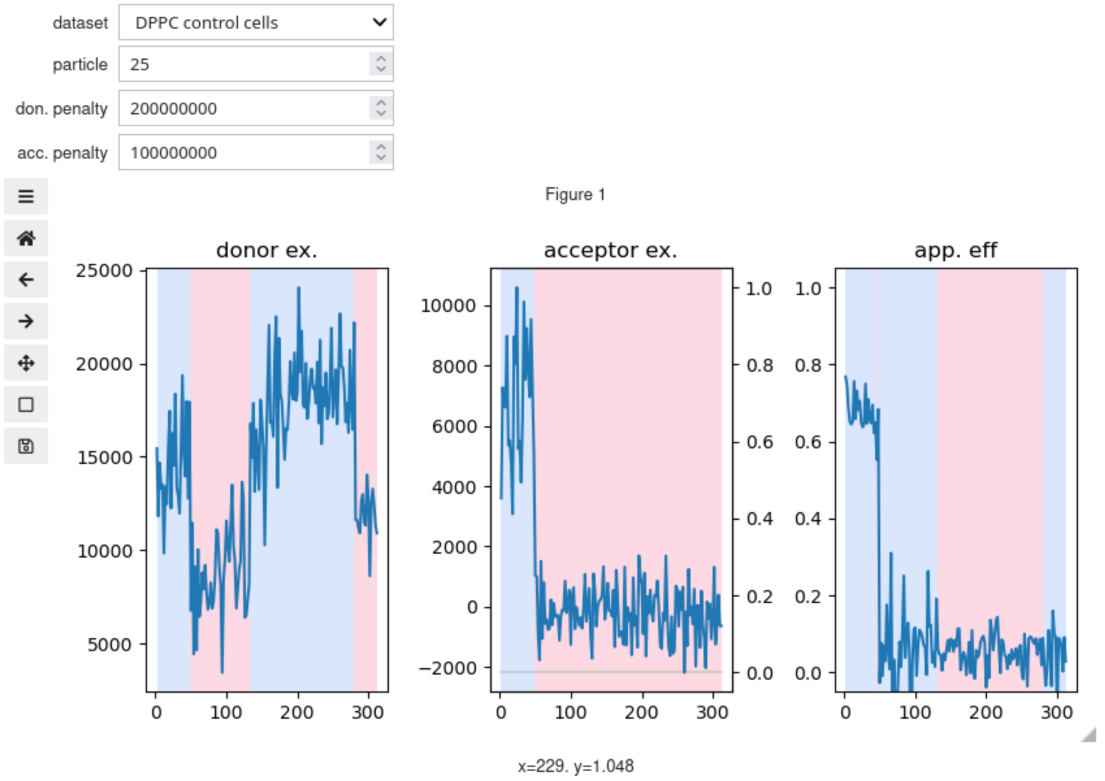
3.10 Calculate apparent FRET efficiencies and stoichiometries

```
[10]: # Calculate apparent FRET values, e.g., efficiency and stoichiometry
      ana.calc_fret_values(a_mass_interp="next")
```

Since acceptor excitation is performed after donor excitation, the acceptor emission upon exceptor excitation at the time of the donor excitation needs to be interpolated, which is specified by `a_mass_interp`. If `a_mass_interp=next`, the algorithm uses the next acceptor excitation frame. If `a_mass_interp=linear`, it performs a linear interpolation. See the code documentation for other interpolation methods. Execute the cell to calculate apparent FRET quantities.

3.11 Find changepoint detection parameters for analysis of bleaching steps

```
[11]: # Find thresholds for changepoint detection in intensity upon donor
      # excitation (d_mass) as well as upon acceptor excitation (a_mass)
      # Typically, one would find first the correct order of magnitude, then
      # fine-tune.
      ana.find_changepoint_options()
```



dataset: DPPC control cells
particle: 25
don. penalty: 200000000
acc. penalty: 100000000

Real particle number: 163

Analyze photobleaching steps in order to (a) identify monomeric probes and (b) accept only datapoints before a photobleaching event in each track. Bleaching steps are found via a changepoint detection alogrithm. Execute `ana.find_changepoint_options()` to display a widget that will allow you to select a track to display, set the penalty changepoint detection parameter individually for the total intensity upon donor and acceptor excitation, and displays the respective time traces for the currently selected track. We suggest first browsing through the tracks until one is found that features one or more bleaching events. Subsequently adjust the penalty for the appropriate channel (total fluorescence intensity upon donor excitation, total fluorescence intensity upon acceptor excitation) so that detected changepoints as indicated by changing background color coincide with the bleaching events. To do so, first find the correct order of magnitude of the penalty (i.e., append zeros to make changepoint detection less sensitive, remove zeros to increase sensitivity), then fine-tune. Continue browsing time traces and refining the parameters until satisfactory results are obtained.

3.12 Perform changepoint detection for analysis of bleaching steps

[12]:

```
# Segment tracks using changepoint detection in the intensity time trace
# upon donor excitation and upon acceptor excitation.
# Manually copy the parameters from above here.
ana.mass_changepoints("donor", penalty=150000000)
ana.mass_changepoints("acceptor", penalty=100000000)
```

ana.mass_changepoints(…) performs changepoint detection on all time traces. It should be called once for donor and once for acceptor excitation. Specify channel name as the first argument and pass the respective penalty which was chosen in the previous stop. Execute the cell.

3.13 Filter tracks according to bleaching steps (optional)

[13]:

```
# Remove trajectories where acceptor does not bleach in a single step and
# donor shows partial bleaching.
# set_bleach_thresh sets maximum intensity which is considered bleached for
# donor and acceptor.
ana.set_bleach_thresh(1000, 500)
ana.filter_bleach_step("donor or acceptor")

if plot_keys:
    dp.dscatter("Bleaching steps")
```

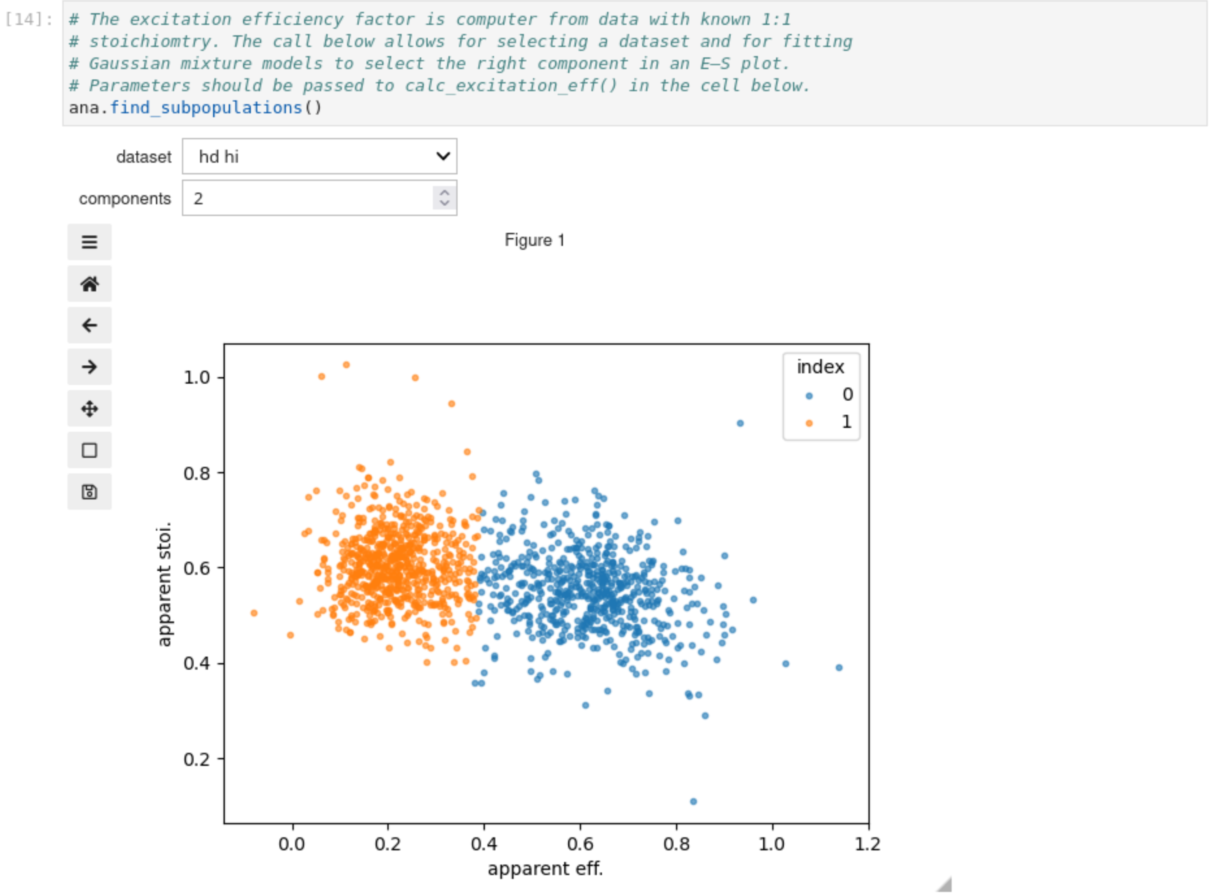
Accept only tracks where there is no partial bleaching by executing ana.filter_bleach_step(…). Additional restrictions can be specified using the condition parameter:

- If condition="acceptor", the acceptor has to bleach in a single step before the donor.
- If condition="donor", the donor has to bleach in a single step before the acceptor.
- If condition="donor or acceptor", one of the fluorophores bleach in a single step.
- If condition="no partial", no additional requirements are enforced.

Use set_bleach_thresh(…) to specify the maximum intensity up to which a signal is considered bleached in the donor (first argument) and acceptor (second argument) channel.

If plot_keys was set in step 3.4, the filtered data will be plotted in the side panel.

3.14 Identify subpopulations for computation of FRET correction factors



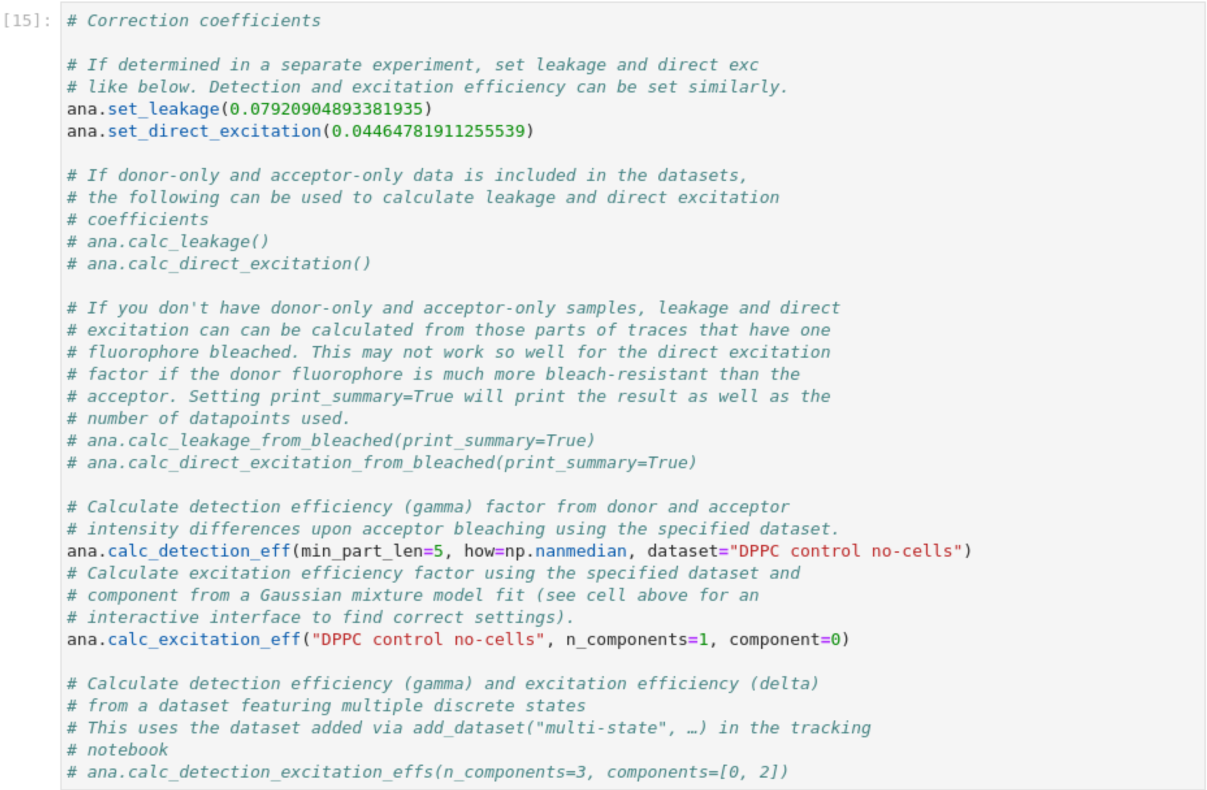
```
[14]: # The excitation efficiency factor is computer from data with known 1:1
      # stoichiomtry. The call below allows for selecting a dataset and for fitting
      # Gaussian mixture models to select the right component in an E–S plot.
      # Parameters should be passed to calc_excitation_eff() in the cell below.
      ana.find_subpopulations()
```

dataset: hd hi

components: 2

Excitation and detection efficiency correction factors can be calculated from a sample with multiple discrete states which manifest themselves via their FRET efficiencies, such as Holliday junctions. In this case, the states have to be identified via a Gaussian mixture fit in the (apparent) efficiency-vs.-stoichiometry plot. If the detection efficiency is calculated from bleaching events, the excitation efficiency can by determined from a subpopulation featuring a $1:1$ donor-to-acceptor ratio, which also has to be identified via a Gaussian mixture fit in the efficiency–vs.–stoichiometry plot. Execute `ana.find_subpopulations()` to display a graphical interface which allows for setting the number of fit components as well as for identifying the subpopulations in the E_{app}–S_{app} plot to use for determination of correction factors.

3.15 Compute FRET correction factors



```
[15]: # Correction coefficients

      # If determined in a separate experiment, set leakage and direct exc
      # like below. Detection and excitation efficiency can be set similarly.
      ana.set_leakage(0.07920904893381935)
      ana.set_direct_excitation(0.04464781911255539)

      # If donor-only and acceptor-only data is included in the datasets,
      # the following can be used to calculate leakage and direct excitation
      # coefficients
      # ana.calc_leakage()
      # ana.calc_direct_excitation()

      # If you don't have donor-only and acceptor-only samples, leakage and direct
      # excitation can can be calculated from those parts of traces that have one
      # fluorophore bleached. This may not work so well for the direct excitation
      # factor if the donor fluorophore is much more bleach-resistant than the
      # acceptor. Setting print_summary=True will print the result as well as the
      # number of datapoints used.
      # ana.calc_leakage_from_bleached(print_summary=True)
      # ana.calc_direct_excitation_from_bleached(print_summary=True)

      # Calculate detection efficiency (gamma) factor from donor and acceptor
      # intensity differences upon acceptor bleaching using the specified dataset.
      ana.calc_detection_eff(min_part_len=5, how=np.nanmedian, dataset="DPPC control no-cells")
      # Calculate excitation efficiency factor using the specified dataset and
      # component from a Gaussian mixture model fit (see cell above for an
      # interactive interface to find correct settings).
      ana.calc_excitation_eff("DPPC control no-cells", n_components=1, component=0)

      # Calculate detection efficiency (gamma) and excitation efficiency (delta)
      # from a dataset featuring multiple discrete states
      # This uses the dataset added via add_dataset("multi-state", …) in the tracking
      # notebook
      # ana.calc_detection_excitation_effs(n_components=3, components=[0, 2])
```

Correction factors can be set or calculated in various ways.

- To set them directly, for instance because they have been determined in a separate experiment, use `ana.set_leakage(…)`, `ana.set_direct_excitation(…)`, `ana.set_detection_eff(…)`, and `ana.set_excitation_eff(…)`, and pass the respective value as the sole parameter.
- If a sample lacking acceptor fluorophores was recorded, `ana.calc_leakage()` can be used. The corresponding donor-only sample had to be added before localization and tracking (see step 1.2).
- Leakage can also be calculated from regular samples' datapoints where the acceptor has been bleached by calling `ana.calc_leakage_from_bleached(…)`.
- Direct excitation can be computed using `ana.calc_direct_excitation()` by analyzing an acceptor-only sample (see also step 1.2) or via `ana.calc_direct_excitation_from_bleached(…)` by analyzing regular samples' datapoints featuring photobleached donors.
- If probes in a dataset feature high FRET efficiencies which remain approximately constant over time, acceptor bleaching can be analyzed to determine the correction factor for detection efficiencies using `ana.calc_detection_eff(…)`. This takes parameters `min_part_length`—the minimum number of datapoints before and after a bleaching step to be analyzed—, `how`–a function that computes a final factor from the individual tracks' values, typically `np.nanmedian` or `np.nanmean`, and `dataset`—the identifier of the dataset to use. Subsequently, a call to `ana.calc_excitation_eff(…)` determines the excitation efficiency correction. As arguments, pass the dataset identifier, the number of components (`n_components`) and the index of the component corresponding to the population with a stoichiometry of 0.5 (`component`) as determined in the widget from the previous step.
- If there are no datasets that allow for calculation of the detection efficiency correction from acceptor bleaching, a sample featuring multiple FRET efficiencies at stoichiometry 0.5 (such as isolated Holliday junctions, which fluctuate between two states) can be used for determination of detection and excitation efficiency corrections by means of the `ana.calc_detection_excitation_effs(…)` call. As arguments, pass the number of components (`n_components`) and a list of component indices to use for fitting (`components`) as determined in the widget from the previous step. If `dataset` is not specified, the dataset specified via `Tracker.add_special_dataset("multi-state", …)` is used.

Choose whatever is appropriate in your case to set the four correction factors and execute the cell.

3.16 Apply FRET correction factors

```
[16]: # Apply correction factors to calculate real FRET efficiencies and
      # stoichiometries
      ana.fret_correction()
```

Execute `ana.fret_correction()` to accurately determine the FRET efficiency and stoichiometry for each detected signal using the correction factors calculated in the previous step.

3.17 Select data before bleaching events

```
[17]: # Select only data before bleaching of a fluorophore, i.e., where the segment
      # number of both donor excitation and acceptor excitation intensity time traces
      # is 0. Select only data recorded during donor ("d") excitation.
      ana.query("fret_d_seg == 0 and fret_a_seg == 0 and fret_exc_type == 'd'")
      # Select only traces where > 75% of datapoints are within reasonable
      # stoichiometry limits
      ana.query_particles("0.35 <= fret_stoi <= 0.6", min_rel=0.75)
      # Remove data points where donor is gone (should not be necessary due to the
      # above)
      # ana.query("fret_d_mass > fret_d_mass.median() * 0.2")

      if plot_keys:
          dp.dscatter("Before bleaching, correct stoi")
```

Execute the cell for final sanity checks: `ana.query("fret_d_seg == 0 and fret_a_seg == 0 and fret_exc_type == 'd'")` only accepts signals upon donor excitation that appear before the first bleaching event in each track. `ana.query_particles("0.35 <= fret_stoi <= 0.6", min_rel=0.75)` accepts tracks where more than 75% of the datapoints have a stoichiometry between 0.35 and 0.6, which corresponds to a 1 : 1 donor-to-acceptor ratio (optional).

If `plot_keys` was set in step 3.4, the filtered data will be plotted in the side panel.

3.18 Find parameters for image segmentation (optional)

```
[18]: # Display an UI to find thresholding parameters for cell images
      ana.find_segmentation_options()
```

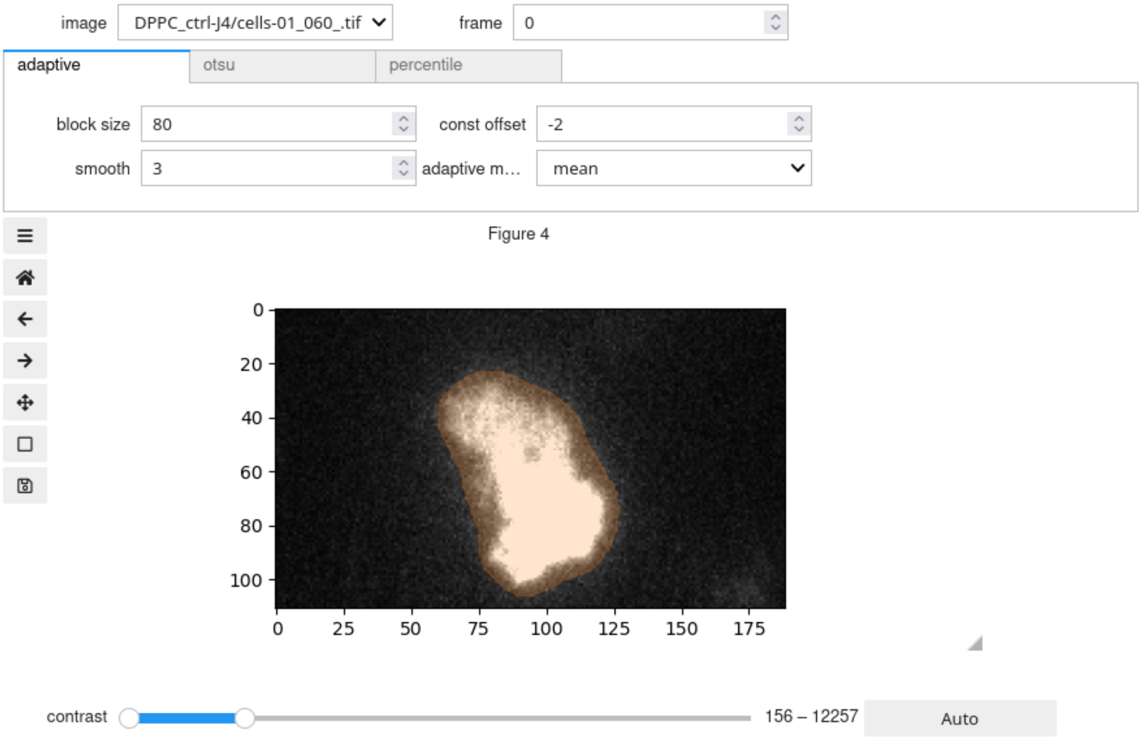


image DPPC_ctrl-J4/cells-01_060_.tif
frame 0

adaptive

block size 80
const offset -2
smooth 3
adaptive m... mean

Figure 4

contrast 156 – 12257

If images for image segmentation were recorded, a thresholding algorithm can be used to automatically determine regions of interest. Execute `ana.find_segmentation_options()` to display a widget designed for finding the appropriate parameters. The tabs allow for choosing the algorithm. In our experience, `adaptive` delivers the best results. Parameters for adaptive thresholding are the block size (`block_size`), σ of a Gaussian filter for smoothing (`smooth`), a constant offset for thresholding (`c`), and the thresholding method (`method`). The proper choice of parameter depends on the imaging conditions, e.g., how cells have been labeled, whether TIRF was used, the excitation intensity which was applied, etc.

3.19 Filter data with results from image segmentation (optional)

```
[19]: # Threshold cell images and select only data within cell-occupied areas
      ana.apply_segmentation([k for k in ana.analyzers if not k.endswith("no-cells")],
                             "adaptive", block_size=65, c=-2, smooth=3, method="mean")

      if plot_keys:
          dp.dscatter("Underneath cells")
```

To restrict certain datasets to datapoints from within regions identified by image segmentation, execute the `ana.apply_segmentation(…)` call. Its first parameter is a list of datasets which should be filtered, such as `["condition1 cells", "condition2 cells"]` or more generically `[k for k in ana.analyzers if "no-cells" not in k]` (all datasets whose names do not contain `"no-cells"`). The second parameter specifies the algorithm to use, e.g. `"adaptive"`. Subsequently, pass the parameters found in the previous step, e.g. `block_size=80, smooth=3, c=-2, method="mean"` as additional arguments to `ana.apply_segmentation(…)`.

If `plot_keys` was set in step 3.4, the filtered data will be plotted in the side panel.

3.20 Save data

```
[20]: # Save results to disk
      ana.save()
```

Save results by executing `ana.save()`. All remarks concerning saving data after tracking (see step 1.10) apply here as well.

3.21 Inspection of data (optional)

Results can be inspected and modified at any point of the analysis. The `Analyzer` instance `ana` holds all data in the `ana.analyzers` Python dictionary, which maps dataset names to `sdt.fret.SmFRETAnalyzer` instances. See the *sdt-python* library documentation for details.

4 Plotting of results and further analysis (`03. Plot.ipynb`)

4.1 Import and setup of required Python packages

```
[1]: %matplotlib inline
     import matplotlib.pyplot as plt
     import numpy as np

     from smfret_analysis import print_info, Plotter
```

Press `Shift+Enter` to execute the cell.

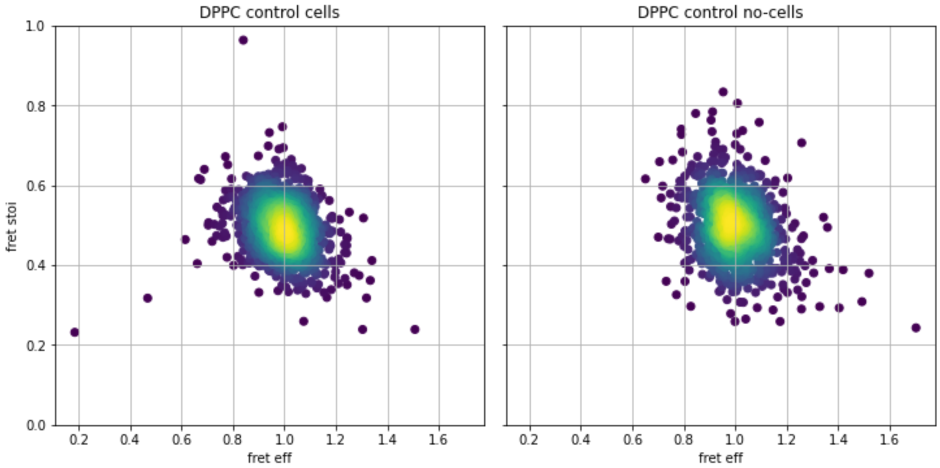
4.2 Load data

```
[2]: p = Plotter()
```

Execute cell to create a `Plotter` instance named `p` This loads the tracking data from save files created by the analysis notebook. The `Plotter()` call take can one optional argument: the file name prefix. Pass whatever was passed to the `save` function in the analysis notebook.

4.3 Generate FRET efficiency-vs.-stoichiometry scatter plots

```
[3]: fig, ax = p.scatter(frame=None, ylim=(0, 1));
     fig.savefig("scatter.pdf", bbox_inches="tight")
```

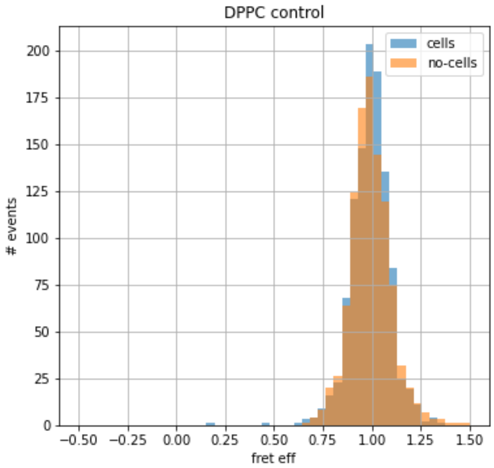


Execute `p.scatter(…)` to draw the plots plots. It is possible to pass `xlim` and `ylim` arguments which set lower and upper limits of x and y axes, respectively. For instance, use `xlim=(-0.5, 1.5)`. If not set, limits are chosen to fit all datapoints. Axis labels can be set via `xlabel` and `ylabel` parameters.

The call returns the *matplotlib* figure instance and array of axes, which can be further manipulated. For example, call `fig.savefig("scatter.pdf")` to save the plot to a file named `scatter.pdf`. `p.scatter(…)` calls `sdt.fret.smfret_scatter(…)`; see the documentation of the *sdt-python* library for additional parameters and details.

4.4 Generate FRET efficiency histograms

```
[4]: fig, ax = p.hist(group_re=(r"(.+) (cells|no-cells)", 1, 2),
                      hist_args={"alpha": 0.6, "density": False});
     fig.savefig("hist.pdf", bbox_inches="tight")
```



Execute `p.hist(…)` to draw the histograms. Arguments described in conjunction with the `p.scatter(…)` method can also be applied here. Furthermore it is possible to specify how to group datasets, i.e., put multiple datasets into the same plot. For instance, this allows to directly compare experimental results with the results from control experiments.

The `group_re` parameter consists of three entries: a regular expression with at least two groups, the index of the group which is used to identify datasets which should be in the same plot, and the index of the group which will be used to identify datasets within individual plots. Note that indices start with 1 in this case. As an example, consider datasets are named `"condition1 cells"`, `"condition1 no-cells"`, `"condition2 cells"`, and `"condition2 no-cells"`. Then, `group_re=[r"(condition.)`

`(cells|no-cells)", 1, 2]` will generate two plots, one for datasets starting with `condition1` and one for datasets starting with `condition2`, since index 1 (i.e., the group `(condition.)`) is specified to identify datasets for different plots. Each plot will contain one `cells` and one `no-cells` histogram, since index 2 (i.e., the group `(cells|no-cells)`) is to be used for datasets within the plots.

The call returns the *matplotlib* figure instance and array of axes, which can be further manipulated. For example, call `fig.savefig("hist.pdf")` to save the plot to a file named `hist.pdf`. `p.hist(…)` calls `sdt.fret.smfret_hist(…)`; see the documentation of the *sdt-python* library for additional parameters and details.

4.5 Further analysis

`p.track_data` is a Python dictionary mapping dataset names to tables (`pandas.DataFrames`) containing single-molecule data, one event per line. These can be exported to various file formats or further analyzed in the Jupyter notebook using, for instance, functionality provided by the *sdt-python* library.

References

1. Python Software Foundation Re — regular expression operations. at <https://docs.python.org/3/library/re.html> (2021).