PyCorrFit
FCS data evaluation
Software Guide

Thomas Weidemann
Max Planck Institute of Biochemistry, Martinsried, Germany

Paul Müller
Biotechnology Center of the TU Dresden, Germany

April 9, 2014

Contents

1 Introduction **3**
1.1 Preface . . . 3
1.2 System prerequisites . . . 3
1.2.1 Hardware . . . 3
1.2.2 Software . . . 4
1.2.3 Running *PyCorrFit* . . . 4
1.3 Workflow . . . 5
1.4 Graphical user interface (GUI) . . . 6

2 The menu bar **8**
2.1 File menu . . . 8
2.1.1 File / Import model . . . 8
2.1.2 File / Load data . . . 8
2.1.3 File / Open session . . . 9
2.1.4 File / Comment session . . . 9
2.1.5 File / Clear session . . . 9
2.1.6 File / Save session . . . 9
2.1.7 File / Exit . . . 9
2.2 Tools menu . . . 9
2.2.1 Tools / Data range . . . 10
2.2.2 Tools / Overlay curves . . . 10
2.2.3 Tools / Batch control . . . 10
2.2.4 Tools / Global fitting . . . 11
2.2.5 Tools / Average data . . . 11
2.2.6 Tools / Trace view . . . 11
2.2.7 Tools / Statistics view . . . 11
2.2.8 Tools / Page info . . . 12
2.2.9 Tools / Slider simulation . . . 12
2.3 Current Page . . . 13
2.3.1 Current Page / Import Data . . . 13
2.3.2 Current Page / Save data (*.csv) . . . 13
2.3.3 Current Page / Save correlation as image . . . 13
2.3.4 Current Page / Save trace view as image . . . 13
2.3.5 Current Page / Close page . . . 13

2.4 Models . . . 14
2.5 Preferences . . . 14
2.6 Help . . . 14

3 Hacker's corner **15**
3.1 External model functions . . . 15
3.2 Internal model functions . . . 16
3.3 Correlation curve file format . . . 16
3.4 New file format . . . 18

4 Theoretical background **18**
4.1 How FCS works . . . 18
4.2 Framework for a single type of fluorescent dye . . . 19
4.3 Autocorrelation of multiple species . . . 21
4.4 Correcting non-correlated background signal . . . 22
4.5 Cross-correlation . . . 22
4.6 Extensions of the method . . . 23
4.6.1 Perpendicular scanning FCS (pSFCS) . . . 23
4.6.2 Circular scanning FCS (cSFCS) . . . 24
4.6.3 Total internal reflection FCS (TIR-FCS) . . . 24
4.7 Fitting . . . 25
4.7.1 Weighted fitting . . . 25
4.7.2 Algorithms . . . 26

5 Implemented model functions **26**
5.1 Confocal FCS . . . 26
5.2 TIR-FCS . . . 28
5.2.1 TIR-FCS with Gaussian-shaped lateral detection volume . . . 29
5.2.2 TIR-FCS with a square-shaped lateral detection volume . . . 30

6 Troubleshooting **32**

Acknowledgements **33**

1 Introduction

1.1 Preface

PyCorrFit emerged from my work in the Schwille Lab[1] at the Biotechnology Center of the TU Dresden in 2011/2012. Since then, the program has been further developed based on numerous input from FCS users, in particular Franziska Thomas, Grzesiek Chwastek, Janine Tittel, and Thomas Weidemann. The program source code is available at GitHub[2]. Please do not hesitate to sign up and add a feature request. If you you find a bug, please let us know via GitHub.

PyCorrFit was written to simplify the work with experimentally obtained correlation curves. These can be processed independently (operating system, location, time). PyCorrFit supports commonly used file formats and enables users to allocate and organize their data in a simple way.

PyCorrFit is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version[3].

What PyCorrFit can do

- Load correlation curves from numerous correlators
- Process these curves
- Fit a model function (many included) to an experimental curve
- Import user defined models for fitting
- Many batch processing features
- Save/load entire *PyCorrFit* sessions
- LaTeX support for data export

What PyCorrFit is not

- A multiple-τ correlator
- A software to operate hardware correlators

1.2 System prerequisites

1.2.1 Hardware

This documentation addresses the processing of correlation curves with *PyCorrFit* and was successfully used with the following setups:

1. APD: Photon Counting Device from PerkinElmer Optoelectronics, Model: `SPCM-CD3017`
   Correlator: Flex02-01D/C from correlator.com with the shipped software `flex02-1dc.exe`.

[1] http://www.biochem.mpg.de/en/rd/schwille/
[2] https://github.com/paulmueller/PyCorrFit
[3] http://www.gnu.org/licenses/gpl.html

2. APD: Photon Counting Device from PerkinElmer Optoelectronics
   Correlator: ALV-6000

3. LSM Confocor2 or Confocor3 setups from Zeiss, Germany.

1.2.2 Software

The latest version of *PyCorrFit* can be obtained from the internet at `http://pycorrfit.craban.de`.

- **MacOSx.** Binary files for Mac OSx 10.6.8 and later are available from the download page.
- **Windows.** For Windows XP and later, stand-alone binary executables are available from the download page.
- **Ubuntu/Debian.** PyCorrFit is available from the Debian repositories and can be installed via the operating systems packaging tool (e.g. `apt-get install pycorrfit`). There are also binaries for Ubuntu at the PyCorrFit download page.
- **Pypi.** The program was written in Python, keeping the concept of cross-platform programming in mind. To run *PyCorrFit* on any other operating system, the installation of Python v.2.7 is required. *PyCorrFit* is included in the package index of `python-pip` (`http://pypi.python.org/pypi/pip`) and can be installed via `pip install pycorrfit`[4].
- **Sources.** You can also directly download the source code at any developmental stage[5]. *PyCorrFit* depends on the following python modules:

  `python-matplotlib` ($\geq$ 1.0.1)
  `python-numpy` ($\geq$ 1.5.1)
  `python-scipy` ($\geq$ 0.8.0)
  `python-sympy` ($\geq$ 0.7.2)
  `python-yaml`
  `python-wxgtk2.8`

LaTeX. *PyCorrFit* can save correlation curves as images using matplotlib. It is also possible to utilize LaTeXto generate these plots. On Windows, installing MiKTeX with "automatic package download" will enable this feature. On MacOSx, the MacTeX distribution can be used. On other systems, a latex distribution, Ghostscript, `dvipng` and the latex packages `texlive-latex-base` and `texlive-math-extra` need to be installed.

1.2.3 Running PyCorrFit

Windows. Download the executable file and double-click on the `PyCorrFit.exe` icon.

Ubuntu/Debian. `PyCorrFit` is integrated into Debian and thus behaves like any other application.

[4] See also the wiki article at `https://github.com/paulmueller/PyCorrFit/wiki/Installation_pip`
[5] See also the wiki article at `https://github.com/paulmueller/PyCorrFit/wiki/Running-from-source`

Mac OSx. When downloading the archive `PyCorrFit.zip`, the binary should be extracted automatically (if not, extract the archive) and you can double-click it to run *PyCorrFit*.If prompted, select *Terminal.app* from the *utilities* folder. Drag the *PyCorrFit* icon onto the Terminal, then MAcOS will assign Terminal.app as the standard application for opening this file (can be checked under properties).

from source. Invoke `python PyCorrFit.py` from the command line.

1.3 Workflow

The following chapter introduces the general idea of how to start and accomplish a fitting project. FCS experiments produce different sets of experimental correlation functions which must be interpreted with appropriate physical models (Chapter 4). Each correlation function refers to a single contiguous signal trace or "run". In *PyCorrFit*, the user must assign a mathematical model function to each correlation function during the loading procedure. The assignment is irreversible in the sense that within an existing *PyCorrFit* session it cannot be changed. This feature assures the stability of the batch processing routine for automated fitting of large data sets. Nevertheless, the fit of different models to the same data can be explored by loading the data twice or simply by creating two different sessions.

Let's briefly discuss a typical example: To determine the diffusion coefficient of a fluorescently labeled protein in free solution, one has to deal with two sets of autocorrelation data: measurements of a diffusion standard (e.g. free dye for which a diffusion coefficient has been published) to calibrate the detection volume and measurements of the protein sample. The protein sample may contain small amounts of slowly diffusing aggregates. While the calibration measurements can be fitted with a one-component diffusion model (T-3D), the protein sample displays two mobility states, monomers and aggregates, which are taken into account by a two-component diffusion model (T-3D-3D). With *PyCorrFit* such a situation can be treated in three ways, having different pros and cons:

1. Create separate sessions for each type of sample and assign different model functions.
2. Assign a one-component model to the dye measurements and a two-component model to the protein measurements when loading consecutively into the same session.
3. Assign a two-component model for all data and, when appropriate, manually inactivate one component by fixing its contribution to 0%.

The first approach is straightforward, however, it requires homogeneous diffusion behavior for each data set. The second strategy has the advantage that the dye and the protein curves, as well as the obtained parameters, can be visually compared during the fitting analysis within the same session. In this case, batch fitting is still possible because it discriminates data sets assigned to different models. In the third case, simultaneous batch fitting is also possible. However, for each dye measurement one has to eliminate the second, slow diffusion species manually, which might be laborious. Inactivating components by fixing parameters is nevertheless a common way to evaluate heterogeneous data sets, for example, a protein sample for which only a subgroup of curves requires a second diffusion component due to occasional appearance of aggregates. Such situations are frequently encountered in intracellular measurements. In conclusion, all three strategies or combinations thereof may be suitable. In any case, the user must decide on model functions beforehand, therefore it is advisable to group the data accordingly.

The fitting itself is usually explored with a representative data set. Here, the user has to decide on starting parameters, the range in which they should be varied, corrections like background, and other fitting options. Once the fit looks good, the chosen settings can be transferred at once to all other pages assigned to the same model using the *Batch control* tool (Section 2.2.3). After flipping through the data for visual inspection one may check the parameters across all pages in the *Statistics view* tool and re-visit outliers (Section 2.2.7). From there, the numerical fit values and example correlation functions can be exported.

1.4 Graphical user interface (GUI)

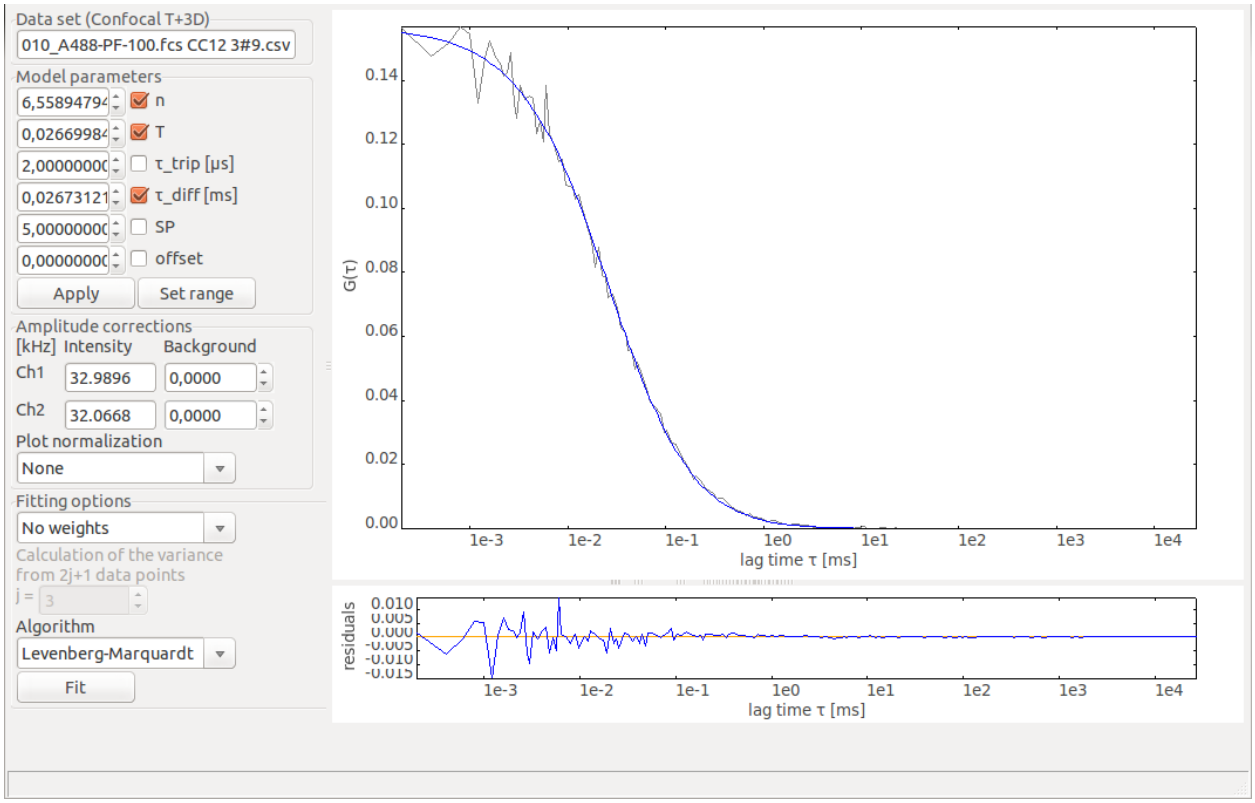

Figure 1, user interface of PyCorrFit: Confocal measurement of nanomolar Alexa488 in aqueous solution. To avoid after-pulsing, the autocorrelation curve was measured by cross-correlating signals from two detection channels using a 50 % beamsplitter. Fitting reveals the average number of observed particles ($n \approx 6$) and their residence time in the detection volume ($\tau_{\mathrm{diff}} = 28\,\mu\mathrm{s}$).

Together with a system's terminal of the platform on which *PyCorrFit* was installed (Windows, Linux, MacOS), the *main window* opens when starting the program as described in Section 1.2.3. The window title bar contains the version of *PyCorrFit* and, if a session was re-opened or saved, the name of the fitting session. A menu bar provides access to many supporting tools and additional information as thoroughly described in Chapter 2.

There are three gateways for experimental data into a pre-existing or a new *PyCorrFit* session (*File/Load data*, *File/Open session*, and *Current page/Import data*). When a session has been opened or correlation data have been loaded, each correlation curve is displayed on a separate page of a notebook. For quick identification of the active data set, a tab specifies the page number, the correlated channels (AC/CC), and the run number in cases where multiple correlation runs are combined in a single file. When clicking a little triangle to the far-right, one can use a drop-down list of all page titles to directly access a particular data

set. Alternatively, the pages can be toggled by tapping the arrow keys (left/right). There can be only one activated page for which the tab appears highlighted.

The active page displaying a correlation function is divided in two panels (Figure 1). At the left hand side the page shows a pile of boxes containing values or fitting options associated with the current model and data set:

- *Data set* specifies the assigned model abbreviation in parentheses and shows a unique identifier for the correlation curve containing the file name, the number of the "run", and the data channel. This string is automatically assembled during the loading procedure (Section 2.1.2). However, during the session it can be manually edited, thereby allowing to re-name or flag certain data during the fitting analysis.

- *Model parameters* displays the values which determine the current shape of the assigned model function (Chapter 4). Initially, starting values are loaded as they were defined in the model description (Section 2.1.1). Little buttons allow a stepwise increase or decrease in units of $1/10^{\text{th}}$. It is also possible to directly enter some numbers. A checkbox is used to set the parameter status to "varied" (checked) or "fixed" (unchecked) during the fitting. At the end, when saving the session, the current set of values together with their indicated names are stored in the *.yaml file (Section 2.1.6).

- *Amplitude corrections* applies additional rescaling to amplitude related parameters like the number of particles n or amplitude fractions associated with different correlation times (n_1, n_2, etc.). Experimental values of non-correlated background intensity can be manually entered for each channel. In addition, the correlation curves can be normalized, to facilitate a visual comparison of the decay.

- *Fitting options* offers weighted fitting (a) and a choice for the fit algorithm (b).

  a) The underlying idea is that data points with higher accuracy should also have a higher impact on model parameters. To derive weights, *PyCorrFit* calculates the variance of the difference between the actual data and a smooth, empiric representation of the curve for a certain neighbourhood. The number of neighbouring data points at each side ($j > 0$) can be set. For such a smooth representation a spline function or the model function with the current parameter set can be used. The default number of knots for the spline function is 5. This number can be manually edited in the dropdown-selector. To view the spline function after each fit, *Preferences/Verbose mode* has to be checked. If the current page is an average, then the standard deviation from the curves that were used to create the average can be used as weights.

  b) Several fitting algorithms can be chosen. We recommend to use the Levenberg-Marquardt algorithm (b). For more information, see Section 4.7.2.

At the right hand side are two graphics windows. The dimensionless correlation functions $G(\tau)$ are plotted against the lag time (τ) on a logarithmic scale. Below, a second window shows the residuals, the actual numerical difference between the correlation data and the model function. Fitting with appropriate models will scatter the residuals symmetrically around zero (x-axis). When weighted fitting was performed, the weighted residuals are shown. A good fit will not leave residuals systematically above or below the x-axis at any time scale.

The main window can be rescaled as a whole to improve data representation. In addition, to zoom in, one can drag a rectangle within the plot area; a double click then restores the

initial scale. Experimental data points are linked by grey lines, the state of the model function is shown in blue. When a weighted fit was applied, the variance of the fit is calculated for each data point and displayed in cyan.

2 The menu bar

The menu bar organizes data management (File), data analysis (Tools), display of correlation functions (Current Page), numerical examples (Model), software settings (Preferences), and software metadata (Help).

2.1 File menu

The File menu organizes the import of theoretical models, experimental correlation data, and opening and saving of entire *PyCorrFit* fitting sessions. However, the numerical fit results are exported from the *Statistics view* panel which can be found under *Tools* (Section 2.2.7).

2.1.1 File / Import model

Correlation data must be fitted to models describing the underlying physical processes which give rise to a particular time dependence and magnitude of the recorded signal fluctuations. Models are mathematical expressions containing parameters with physical meaning, like the molecular brightness or the dwell time through an illuminated volume. While the most commonly used standard functions are built-in, the user can define new expressions. Some examples can be found at GitHub in the *PyCorrFit* repository, e.g. circular scanning FCS [23] (Figure 2) or a combination of diffusion and directed flow [4]. External model functions are discussed in detail in Section 3.1.

2.1.2 File / Load data

Load data is the first way to import multiple correlation data sets into a *PyCorrFit* session. The supported file formats can be found in a drop-down list of supported file endings in the pop-up dialog *Open data files*:

(1) All supported files	default
(2) ALV (*.ASC)	ALV Laser GmbH, Langen, Germany
(3) Correlator.com (*.SIN)	www.correlator.com, USA
(4) Zeiss ConfoCor3 (*.fcs)	AIM 4.2, ZEN 2010, Zeiss, Germany
(5) Matlab ‘Ries (*.mat)	EMBL Heidelberg, Germany
(6) PyCorrFit (*.csv)	Paul Müller, TU Dresden, Germany
(7) PyCorrFit session (*.pcfs)	Paul Müller, TU Dresden, Germany
(8) Zip file (*.zip)	Paul Müller, TU Dresden, Germany

While (2)-(4) are file formats associated with commercial hardware, (5) refers to a MATLAB based FCS evaluation software developed by Jonas Ries in the Schwille lab at TU Dresden, (6) is a text file containing comma-separated values (csv) generated by PyCorrFit via the command *Current Page / Save data*. Zip-files are automatically decompressed and can be imported when matching one of the above mentioned formats. In particular loading of *.pcfs

files (which are actually zip files) is a possibility to re-import correlation data from entire *PyCorrFit* sessions. However, these data are treated as raw, which means that all fitting parameters and model assignments are lost.

During loading, the user is prompted to assign fit models in the *Choose Models* dialogue window. There, curves are sorted according to channel (for example AC1, AC2, CC12, and CC21, as a typical outcome of a dual-color cross-correlation experiment). For each channel a fit model must be selected from the list (see Section 2.4):

If a file format is not yet listed, the correlation data could be converted into a compatible text-file (*.csv) or bundles of *.csv files within a compressed archive *.zip. For reformatting correlation data, have a look at Section 3.3.

2.1.3 File / Open session

This command is the second way to import data into PyCorrFit. In contrast to *Load data*, it opens an entire fitting project, which was previously saved with *PyCorrFit*. Session files are *.zip files named *.pcfs. These files contain all information to restore a session, including comments, model assigned correlation data, and the current state of parameters for each data set (Section 2.1.6).

2.1.4 File / Comment session

This command opens a window to place text messages that can be used to annotate a fitting session.

2.1.5 File / Clear session

This command closes all pages. The user is prompted to save the current session.

2.1.6 File / Save session

In addition to display and fit individual curves, a strong feature of *PyCorrFit* is to save an entire fitting project as a single session. Sessions allow the user to revisit and explore different models, fitting strategies, and data sets. Importantly the work can be saved at any stage.

The number of files bundled in a session varies depending on the number of data sets (pages), the number of used models, and what was done during the fitting. A detailed description can be found in the Readme.txt file attached to each session. For example, the numerical correlation and intensity data are saved separately as *.csv text files. However, in contrast to the *Save data (*.csv)* command of the *Current Page* menu, there are no metadata in the header, just tables containing the numerical values. In sessions, the fitting parameters are stored separately in the human-readable data serialization format *.yaml.

2.1.7 File / Exit

This command closes down *PyCorrFit*. The user is prompted to save the session under the same or a different name.

2.2 Tools menu

The *Tools* menu provides access to a series of accessory panels which extent the capability of the main window. These accessory panels can stay open during the entire analysis. Open

panels appear checked in the menu. Most operations can be executed across the entire data set with a single mouse click.

2.2.1 Tools / Data range

This panel limits the range of lag times which are displayed in the main window panel. At the same time it defines the range of points which are used for fitting. For example, this feature can be applied to remove dominant after-pulsing of the avalanche photo diodes (APDs) which may interfere with Triplet blinking at short lag times. The user has the options to *Apply* the channel settings only to the current page or he can *Apply to all pages*. In contrast to *Batch control*, this operation ignores whether the data are assigned to different models.

Power user, who frequently load and remove data sets, may take advantage of a checkbox to fix the channel selection for all newly loaded data sets.

2.2.2 Tools / Overlay curves

This window displays the correlation data (not the fit curves) of all pages in a single plot. The curves can be discriminated by color. If only one curve is selected, it appears in red. Curves with ambiguous shape can easily be identified, selected, and removed by clicking *Apply*. A warning dialogue lists the pages which will be kept.

Data representation is synchronized with the page display in the *Main window*. For example, narrowing the range of lag times by *Data range* is immediately updated in the *Overlay curves* tool. Likewise, their normalization of the amplitudes to unity.

The other way round, some tools directly respond to the selections made in the *Overlay curves* tool: *Global fitting*, *Average curves*, and *Statistics view* allow to perform operations on an arbitrary selection of pages which can be specified by page number. Instead of manually typing their numbers, the curves may be selected within the *Overlay curves* tool. The respective input fields are immediately updated.

The tool is closed by the button *Cancel*. All the listed data sets will be kept. However, the selections transferred to the *Global fitting*, *Average curves*, and *Statistics view* tools are kept as well.

2.2.3 Tools / Batch control

By default, the current page is taken as a reference to perform automated fitting. A batch is defined as the ensemble of correlation data sets (pages) assigned to the same model function within a session. A session can therefore have several batches, even for the same data.

For fitting, it is crucial to carefully define the starting parameters, whether parameters should be fixed or varied, the range of values which make physically sense, and other options offered within the *Main window*. By executing *Apply to applicable pages*, these settings are transferred to all other pages assigned to the same fit model. Note that this includes the range of lag times (lag time channels) which may have been changed with the *Data range* tool for individual pages.

The button *Fit applicable pages* then performs fitting on all pages of the same batch. Alternatively, the user can define an external source of parameters as a reference, i.e. the first page of some *Other session* (*.pcfs). However, this assumes a consistent assignment of model functions.

2.2.4 Tools / Global fitting

Global fitting is useful when experimental curves share the same values for certain physical parameters. For example, due to physical constraints in two-focus FCS both autocorrelation curves and cross-correlation curves should adopt the same values for the diffusion time τ_{diff} and the number of particles n. A global fit can be applied such that n and τ_{diff} are identical for all data sets. All curves are added to a single array. In contrast to fixing the shared parameters across a batch, in *Global fitting* χ^2 is minimized for all data sets simultaneously (see Section 4.7). To perform *Global fitting*, a subset of curves has to be selected by typing the numbers into the input field or by highlighting the pages via the *Overlay curves* tool.

2.2.5 Tools / Average data

Often in FCS, the measurement time at a particular spot is divided in several runs. This approach is taken when occasional, global intensity changes are superimposed on the molecular fluctuations of interest. Then the user has to sort out the bad runs. After fitting, one may want to re-combine the data to export a cleaned average correlation function. This can be done with the tool *Average data*, for which a subset of curves has to be selected by typing the numbers into the input field or by highlighting the pages via the *Overlay curves* tool. For averaging, there are constraints:

1. Since the correlation curves are averaged point by point this requires the same number of lag time channels. Runs of different length cannot be averaged.

2. The tool can only average data sets which are exclusively autocorrelation or cross-correlation.

3. The user can check a box to enforce the program to ask for data sets with the same model as the current page. This may help to avoid mistakes when selecting pages.

The averaged curve is shown on a separate page. The new *Filename/title* receives the entry *Average [numbers of pages]*. The assigned model is by default the same as for the individual pages. However, while averaging, the user can choose a different model from a drop-down list.

2.2.6 Tools / Trace view

FCS theory makes assumptions about the thermodynamic state of the system. Signal fluctuations can only be analyzed when the system is at equilibrium or at a sufficiently stable steady state. Global instabilities on the time scale of the measurement itself, e.g. photobleaching, have dramatic effect on the shape of the measured correlation curve. Therefore, it is common practice to check the correlated intensity trace for each curve. Trace view simply displays the signal trace for each correlation function. The window stays open during the session and can be used to revisit and flag ambiguous data sets.

2.2.7 Tools / Statistics view

The goal of a correlation analysis is to determine experimental parameter values with sufficient statistical significance. However, especially for large data sets, it can get quite laborious to check all of the individual values on each page. We designed the *Statistics view* panel to review the state of parameters across the experimental batch (pages assigned to the same model) in a single plot, thereby facilitating to the identification of outliers.

The current page is taken as a reference for the type of model parameters which can be displayed. The user can choose different *Plot parameters* from a drop-down list. A subset of pages within the batch can be explicitly defined by typing the page numbers into the input field or by highlighting in the *Overlay curves* tool. Note that page numbers which refer to different models than the current page are ignored.

The *Statistics view* panel contains a separate *Export* box, where parameters can be selected (checked) and saved as a comma separated text file (*.csv). Only selected page numbers are included.

2.2.8 Tools / Page info

Page info is a most verbose summary of a data set. The panel *Page info* is synchronized with the current page. The following fields are listed:

1. Version of *PyCorrFit*
2. Field values from the main window (filename/title, model specifications, page number, type of correlation, normalizations)
3. Actual parameter values (as contained in the model function)
4. Supplementary parameters (intensity, counts per particle, duration, background correction, etc.)
5. Fitting related information (Chi-square, channel selection, varied fit parameters) .
6. Model doc string (Section 2.4)

The content of Page info is saved as a header when exporting correlation functions via the command *Current page / Save data (*.csv)* (Section 2.3.2).

2.2.9 Tools / Slider simulation

This tool visualizes the impact of model parameters on the shape of the model function of a current page. Such insight may be useful to choose proper starting values for fitting or to develop new model functions. For example, in the case of two parameters that trade during the fitting one may explore to which extent a change in both values produces similar trends.

Two variables (A and B) have to be assigned from a drop-down list of parameters associated with the current model function. For each of these, the *Slider simulation* panel shows initially the starting value (x) as a middle position of a certain range (from 0.1*x to 1.9*x). The accessible range can be manually edited and the actual value of the slider position is displayed at the right hand side of the panel. Dragging the slider to lower (left) or higher (right) values changes the entry in the box *Model parameters* of the *Main window* and accordingly the shape of the model function in the plot. By default the checkbox *Vary A and B* is active meaning that both variables during *Slider simulation* can be varied independently.

In addition, the variables A and B can be linked by a mathematical relation. For this a mathematical operator can be selected from a small list and the option *Fix relation* must be checked. Then, the variable B appears inactivated (greyed out) and the new variable combining values for A and B can be explored by dragging.

2.3 Current Page

This menu compiles import and export operations referring exclusively to the active page in the main window.

2.3.1 Current Page / Import Data

This command is the third way to import data into a pre-existing session. Single files containing correlation data can be imported as long as they have the right format (Section 2.1.2). In contrast to *Load data* from the *File* menu, the model assignment and the state of the parameters remains. The purpose of this command is to compare different data sets to the very same model function for a given set of parameters. After successful import, the previous correlation data of this page are lost.

To avoid this loss, one could first generate a new page via the menu *Models* (Section 2.4), select a model function and import data there. This is also a possibility to assign the very same data to different models within the same session.

2.3.2 Current Page / Save data (*.csv)

For the documentation with graphics software of choice, correlation curves can be exported as a comma-separated table. A saved *PyCorrFit* text-file (*.csv) will contain a hashed header with metadata from the *Page info* tool (Section 2.2.8), followed by the correlation and fitting values in tab-separated columns: *Channel (tau [s])*, *Experimental correlation*, *Fitted correlation*, *Residuals*, and *Weights (fit)*.

Below the columns, there are again 5 rows of hashed comments followed by the intensity data in two columns: *Time [s]* and *Intensity trace [kHz]*. Note that there are no assemblies of "multiple runs", since *PyCorrFit* treats these as individual correlation functions. A *.csv file therefore contains only a single fitted correlation curve and one intensity trace for autocorrelation or two intensity traces for cross-correlation.

2.3.3 Current Page / Save correlation as image

The correlation curve can be exported as bitmap (e.g. *.png for quick documentation) or as a scalable vector graphic (e.g. *.pdf for post-processing in *Adobe Illustrator* or *Inkscape*). The plot contains a legend and fitting parameters. Note that the variable τ (= tau) cannot be displayed using Unicode with Windows. A LaTeXformatted image can be exported when the option *Use Latex* is checked in the *Preferences* menu (Section 2.5). Furthermore, if *Preferences/Verbose mode* is checked, the plot can be edited before saving[6].

2.3.4 Current Page / Save trace view as image

An image of the trace can be exported in the same way as for the correlation curve (see above).

2.3.5 Current Page / Close page

Closes the page; the data set is removed from the session. The page numbers of all other pages remain the same. The command is equivalent with the closer (x) in the tab.

[6]The plots are generated with matplotlib, http://matplotlib.org/

2.4 Models

When choosing a model from the *Models* menu, a new page opens and the model function is plotted according to the set of starting values for parameters as they were defined in the model description. The lists contains all of the implemented model functions, which can be selected during *File / Load data*. The parameters can be manipulated to explore different shapes; the tool *Slider simulation* can also be used. Via *Current page / Import data*, the model may then be fitted to an experimental data set. Standard model functions for a confocal setup are (Section 5.1):

- Confocal (Gaussian): T+3D Triplet blinking and 3D diffusion
- Confocal (Gaussian): T+3D+3D Triplet with two diffusive components
- Confocal (Gaussian): T-2D Triplet blinking and 2D diffusion
- Confocal (Gaussian): T-2D-2D Triplet with two diffusive components
- Confocal (Gaussian): T-3D-2D Triplet with mixed 3D and 2D diffusion

There is also a collection of models for FCS setups with TIR excitation (Section 5.2):

- TIR (Gaussian/Exp.): 3D 3D diffusion
- TIR (Gaussian/Exp.): T+3D+3D Triplet with two diffusive components
- TIR (Gaussian/Exp.): T+3D+2D Triplet with mixed 3D and 2D diffusion

In addition, there are may be user defined model functions which have been imported previously via *File / Import model* (Section 2.1.1).

2.5 Preferences

Use Latex If the user has a Tex distribution installed (e.g. MikTex for Windows), checking this option will generate LaTeXformatted plots via the *Current page / Save [...] as image* commands.

Verbose mode If checked, this will cause the *PyCorrFit* to display graphs that would be hidden otherwise. In weighted fitting with a spline, the spline function used for calculating the weights for each data points is displayed[7]. When saving the correlation curve as an image (Section 2.3.3), the plot will be displayed instead of saved. If "Use Latex" is checked, these plots will also be Latex-formatted. The advantage in displaying plots is the ability to zoom or rescale the plot from within *PyCorrFit*.

Show weights Checking this option will visualize the weights for each data point of the correlation function in the plot, as well as in the exported image. Note that the weights are always exported when using the *Save data (*.csv)* command from the *Current page* menu.

2.6 Help

Documentation. This entry displays this documentation using the systems default PDF viewer.

[7] For obvious reasons, such a plot is not generated when using the iteratively improved *Model function* or the actual *Average* correlation curve for weighted fitting.

Wiki. This entry displays the wiki of *PyCorrFit* on *GitHub*. Everyone who registers with *GitHub* will be able to make additions and modifications. The wiki is intended for end-users of *PyCorrFit* to share protocols or to add other useful information.

Update establishes a link to the *GitHub* website to check for a new release; it also provides a few web links associated with *PyCorrFit*

Shell. This gives Shell-access to the functions of *PyCorrFit*. It is particularly useful for trouble-shooting.

Software. This lists the exact version of *Python* and the corresponding modules with which *PyCorrFit* is currently running.

About. Information of the participating developers, the license, and documentation writers.

3 Hacker's corner

3.1 External model functions

PyCorrFit supports the import of your own model functions. If your model function is not implemented, i.e. not available in the *Models* menu, writing a short text file containing a model description is the easiest way to make *PyCorrFit* work. Some examples can be found at GitHub in the *PyCorrFit* repository, e.g. circular scanning FCS [23] (Figure 2) or a combination of diffusion and directed flow [4]. Model functions are imported as text files (*.txt) that must follow a certain format and syntax:

- **Encoding**: *PyCorrFit* can interpret the standard Unicode character set. The model files have to be encoded in *UTF-8*.
- **Comments**: Lines starting with a hash (`#`), empty lines, or lines containing only white space characters are ignored. The only exception is the first line starting with a hash followed by a white space and a short name of the model. This line is evaluated to complement the list of models in the dialogue *Choose model*, when loading the data. Imported models are available from the menu *Models/User*.
- **Units**: *PyCorrFit* works with internal units for:
  - Time: 1 ms
  - Distance: 100 nm
  - Diffusion coefficient: $10\,\mu\mathrm{m}^2\mathrm{s}^{-1}$
  - Inverse time: $1000\,\mathrm{s}^{-1}$
  - Inverse area: $100\,\mu\mathrm{m}^{-2}$
  - Inverse volume: $1000\,\mu\mathrm{m}^{-3}$
- **Parameters:** To define a new model function, new parameters can be introduced. Parameters are defined by a sequence of strings separated by white spaces containing name, the dimension in angular brackets, the equal sign, and a starting value which appears in the main window for fitting. For example: `D [10 µm²s⁻¹] = 5.0`. User defined dimensions are only for display; thus mathematical expressions must correctly

account for their conversion from internal units of *PyCorrFit*. The parameter names contain only alphabetical (not numerical) characters. `G`, variables starting with `g`, as well as the numbers `e` and `pi` are already mapped and cannot be used freely.

- **Placeholder:** When defining composite mathematical expressions for correlation functions, one can use place-holders. Place-holders start with a lower-case 'g'. For example, the standard, Gaussian 3D diffusion in free solution may be written as
  - `gTrp = 1+ T/(1-T)*exp(-tau/tautrip)`
  - `gTwoD = 1/(1+tau/taudiff)`
  - `gThrD = 1/sqrt(1+tau/(taudiff*S**2))`

The individual parts are then combined in the last line of the *.txt file, where the correlation function is defined starting with upper-case 'G':

```
G = 1/n * gTrp * gTwoD * gThrD
```

For reference of mathematical operators, check for example www.tutorialspoint.com / python / python_basic_operators.htm. To illustrate a more complex example, a model function for circular scanning FCS is shown in Figure 2. External models will be imported with internal model function IDs starting at 7000. Models are checked upon import by the Python module sympy. If the import fails, there is most likely syntax error in the model file.

3.2 Internal model functions

Alternatively, new models can be implemented by programming of the models module of *PyCorrFit*. First, edit the code for `__init__.py` and then add the script containing the model function. There is no Tutorial yet, but the implemented model files are self-explanatory. If you need help creating a new internal model function and/or want to publish it with *PyCorrFit*, do not hesitate to contact an active developer by creating a new issue on GitHub.

3.3 Correlation curve file format

PyCorrFit can read correlation data from many file formats. If a file format is not yet listed, the correlation data could be converted into a compatible text-file (*.csv) or bundles of *.csv files within a compressed *.zip archive. For reformatting the following requirements must be fulfilled:

- **Encoding**: *PyCorrFit* uses the standard Unicode character set (UTF-8). However, since no special characters are needed to save experimental data, other encodings may also work. New line characters are `\r\n` (Windows).
- **Comments**: Lines starting with a hash (`#`), empty lines, or lines containing only white space characters are ignored. Exceptions are the keywords listed below.
- **Units**: PyCorrFit works with units/values for:
  - Time: 1 ms
  - Intensity: 1 kHz
  - Amplitude offset: $G(0) = 0$ (not 1)

```
# CS-FCS T+2D+2D+S (Confocal)

# Circular Scanning FCS model function for two 2D-diffusing species
# including triplet component.

## Definition of parameters:
# First, the parameters and their starting values for the model function
# need to be defined. If the parameter has a unit of measurement, then it
# may be added separated by a white space before the "=" sign. The starting
# value should be a floating point number. Floating point abbreviations
# like "1e-3" instead of "0.001" may be used.

# Diffusion coefficient of first component
D1 [10μm²/s] = 200.0
# Diffusion coefficient of second component
D2 [10μm²/s] = 20.0
# Fraction of species one
F1 = 1.0
# Half waist of the lateral detection area (w0 = 2*a)
a [100nm] = 1.0
# Particle number
n = 5.0
# Scan radius
R [100nm] = 3.850
# Frequency
f [kHz] = .2
# Triplet fraction
T = 0.1
# Triplet time
tautrip [ms] = 0.001
offset = 0.00001

# The user may wish to substitute certain parts of the correlation function
# with other values to keep the formula simple. This can be done by using
# the prefix "g". All common mathematical functions, such as "sqrt()" or
# "exp()" may be used. For convenience, "pi" and "e" are available as well.

gTriplet = 1. + T/(1-T)*exp(-tau/tautrip)
gScan1 = exp(-(R*sin(pi*f*tau))**2/(a**2+D1*tau))
gScan2 = exp(-(R*sin(pi*f*tau))**2/(a**2+D2*tau))
gTwoD1 = F1/(1.+D1*tau/a**2)
gTwoD2 = (1-F1)/(1.+D2*tau/a**2)

# The final line with the correlation function should start with a "G"
# before the "=" sign.

G = offset +  1./n * (gTwoD1 * gScan1 + gTwoD2 * gScan2) * gTriplet
```

Figure 2, user defined model function for PyCorrFit: The working example shows a model function for circular scanning FCS (see also its appearance in the *Main window* Figure 3

- **Keywords:**[8] *PyCorrFit* reads the first two columns containing numerical values. The first table (non-hashed) is recognized as the correlation data containing the lag times in the first and the correlation data in the second column. (In case the *.csv file has been generated with *PyCorrFit* up to three additional columns containing the fit function are ignored). The table ends, when the keyword `# BEGIN TRACE` appears. Below this line the time and the signal values should be contained in the first two columns. If cross-correlation data have to be imported a second trace can be entered after the keyword `# BEGIN SECOND TRACE`.

- **Tags:**[9] Channel information can be entered using defined syntax in a header. The keyword

  ```
  # Type AC/CC Autocorrelation
  ```

  assigns the tag `AC` and the keyword

  ```
  # Type AC/CC Crosscorrelation
  ```

  assigns the tag `CC` to the correlation curve. These strings are consistently displayed in the user interface of the respective data page in *PyCorrFit*. If no data type is specified, autocorrelation is assumed. Tags may be specified with additional information like channel numbers, e.g.

  ```
  # Type AC/CC Autocorrelation _01.
  ```

  In this case the tag `AC_01` is generated. This feature is useful to keep track of the type of curve during the fitting and when post-processing the numerical fit results.

3.4 New file format

Alternatively, new file formats can be implemented by programming of the readfiles module of *PyCorrFit*. First, edit the code for `__init__.py` and then add the script `read_FileFormat.py`. There is no Tutorial yet, but the implemented scripts are self-explanatory. If you need help implementing a new file format and/or want to publish it with *PyCorrFit*, do not hesitate to contact an active developer by creating a new issue on GitHub.

4 Theoretical background

In the first place, *PyCorrFit* was designed to evaluate FCS data, therefore we focus on fluorescence. However, the correlation theory could be applied to any other stochastic signal.

4.1 How FCS works

FCS is a method to determine molecular properties of stochastically moving fluorescent molecules in solutions [6, 15, 16]. The solution may be a liquid volume (3D) or a lipid membrane (2D) [13, 32, 42]. The diffusion may be free or anomalous due to barriers or obstacles [37, 41]. The size of the diffusing particles range from synthetic dyes (800 Da) to large complexes or aggregates of labelled macromolecules (several MDa). Particles which differ in size or emission behaviour can be discriminated. Therefore FCS is a powerful tool to study molecular recognition [2, 11].

The measurement principle is to illuminate a small open volume within solutions of fluorescent molecules and to detect their molecular transits with a sub-microsecond time resolution

[8] Keywords are case-insensitive.
[9] Tags are case-insensitive.

by sensitive optics. The stochastic movements and other processes affecting fluorescence emission generate a fluctuating signal. The typical time pattern of these fluctuations can be revealed by a correlation analysis. The shape and time range of the decaying correlation function is defined by the exact geometry of the FCS detection volume in conjunction with the properties of the molecular system at hand. To derive hard numbers, the system must be parametrized as a theoretical model. Once the geometry of the detection volume is characterized, parameters can be determined by comparing the model function with experimental data (mathematical fitting, Figure 1).

4.2 Framework for a single type of fluorescent dye

The following equations are described in many papers in different ways. Here we follow the notation of one of us [38]. Fluorescence signals are a result of absorption and emission of photons by a fluorophore. Under experimental conditions, the signal depends on the time dependent distribution of fluorescent particles in the sample volume $c(\vec{r},t)$ and the normalized instrumental detection efficiency $W(\vec{r})$. The total intensity, signal $S(t)$, is the sum of all photons emitted from fluorophores at different positions within the detection volume:

$$S(t) = q \int W(\vec{r}) c(\vec{r},t)\, dV \tag{1}$$

The factor q combines all the photo-physical quantities associated with fluorescence emission like absorption cross section, quantum yield, and the peak intensity of excitation (laser power). In the following, time averages of observables are indicated by angular brackets. We apply the ergodic theorem (see below).

$$\langle S(t)\rangle = \lim_{t\to\infty} \int S(t)\, dt = q \int W(\vec{r}) c\, dV = qn \tag{2}$$

Equation 2 reveals that q is the instrument dependent molecular brightness (kHz/particle), i.e. the average signal divided by the average number of particles n observed within the effective detection volume $V_{\text{eff}} = \int W(\vec{r})\, dV$. During FCS measurements the detected signal is correlated by computing a normalized autocorrelation function:

$$G(\tau) = \frac{\langle S(t)\cdot S(t+\tau)\rangle}{\langle S(t)\rangle^2} - 1 = \frac{\langle \delta S(t)\cdot \delta S(t+\tau)\rangle}{\langle S(t)\rangle^2} = \frac{g(\tau)}{\langle S(t)\rangle^2} \tag{3}$$

Here, τ denotes the lag time used for correlation and $\delta S(t) = S(t) - \langle S\rangle$ the amplitude of the signal fluctuation for a given time point. Equation 3 defines a function with a finite intercept decaying to zero, whereas $g(\tau)$, the non-normalized correlation function, decays to a finite value $\langle S\rangle^{-2}$. To visualize correlation times, the functions $G(\tau)$ are typically plotted against $\log(\tau)$ (Figure 1).

A general way to derive theoretical model functions is to evaluate Equation 3 with explicit expressions describing the instrumental detection efficiency $W(\vec{r})$ and the molecular dynamics governing the local fluorophore concentrations $c(\vec{r},t)$. For example, free diffusion of the molecules can be described by the so-called diffusion propagator.

$$P_{\text{d}}\left(\vec{r}\,'|\vec{r},\tau\right) = \frac{1}{\left(4\pi D_t \tau\right)^{3/2}} \exp\left[-\frac{|\vec{r}\,' - \vec{r}\,|}{4D_t\tau}\right] \tag{4}$$

The propagator $P_{\text{d}}\left(\vec{r}\,'|\vec{r},\tau\right)$ is the conditional probability of a particle with diffusion coefficient D_t to move from $\vec{r}$ to $\vec{r}\,'$ within a time period τ. The probability to find a particle

inside the volume d^3r is simply the ratio of volumes d^3r/V. Such a diffusion propagator leads to a Gaussian shaped probability distribution spreading in time when the molecules successively roam the sample volume V. Assuming that the time average and the ensemble average are equivalent (ergodic theorem), the average signal of a single particle is its molecular brightness q normalized to its contribution to the entire volume, because the integral over the detection efficiency function is exactly the effective measurement volume V_{eff}.

$$\langle s(t)\rangle = q\frac{\int W(\vec{r})\,dV}{V} = \frac{V_{\text{eff}}}{V}q \tag{5}$$

Accordingly, we can express the average product of two signals separated by τ as

$$\langle s(t)\cdot s(t+\tau)\rangle = \frac{q^2}{V}\iint W(\vec{r})P_{\text{d}}\left(\vec{r}'|\vec{r},\tau\right)W(\vec{r}')dVdV' \tag{6}$$

With the definition of the signal $S(t)$ and its average $\langle S(t)\rangle$

$$S(t) = \Sigma_{k=1}^{n} s_k(t) \tag{7}$$

$$\langle S(t)\rangle = \Sigma_{k=1}^{n}\langle s_k(t)\rangle := N\langle s(t)\rangle \tag{8}$$

- where N is the total number of particles in the sample volume V - we can reduce $G(\tau)$ to the sum of individual molecular contributions

$$\begin{aligned} G(\tau) = \frac{\langle S(t)\cdot S(t+\tau)\rangle}{\langle S(t)\rangle^2} - 1 &= \frac{\overbrace{N\langle s(t)\cdot s(t+\tau)\rangle}^{\text{same particles}} + \overbrace{N(N-1)\langle s(t)\rangle^2}^{\text{different particles}}}{N^2\langle s(t)\rangle^2} - 1 \\ &\approx \frac{\langle s(t)\cdot s(t+\tau)\rangle}{N\langle s(t)\rangle^2} \end{aligned} \tag{9}$$

Inserting Equation 5 and Equation 6 yields

$$G(\tau) = \frac{V}{N}\frac{\iint W(\vec{r})P_{\text{d}}\left(\vec{r}'|\vec{r},\tau\right)W(\vec{r}')dVdV'}{\left(\int W(\vec{r})\,dV\right)^2} \tag{10}$$

Note that the molecular brightness q cancels; when considering multiple species with different q, this is no longer the case. Solving these integrals for the confocal detection scheme yields a relatively simple equation containing the diffusion coefficient D_t (molecular property) and the $1/e^2$ decay lengths w_0 and z_0 capturing the dimension of the Gaussian detection volume transversal and parallel to the optical axis, respectively (instrumental properties).

$$G(\tau) = \frac{1}{n}\left(1+\frac{4D_t\tau}{w_0^2}\right)^{-1}\left(1+\frac{4D_t\tau}{z_0^2}\right)^{-1/2} \tag{11}$$

The inverse intercept $(G(0))^{-1}$ is proportional to the total concentration of oberved particles $C = N/V = n/V_{\text{eff}} = n/(\pi^{3/2}w_0^2z_0)$. It is common to define the diffusion time $\tau_{\text{diff}} = {w_0}^2/4D_t$ and the structural parameter $SP = z_0^2/w_0^2$ as a measure of the elongated detection volume. Replacement finally yields the well known autocorrelation function for 3D diffusion in a confocal setup (Model ID 6012)

$$G(\tau) \overset{\text{def}}{=} G^{\text{D}}(\tau) = \frac{1}{n}\overbrace{\left(1+\frac{\tau}{\tau_{\text{diff}}}\right)^{-1}}^{\text{2D}}\overbrace{\left(1+\frac{\tau}{SP^2\,\tau_{\text{diff}}}\right)^{-1/2}}^{\text{3D}} \tag{12}$$

For confocal FCS, both the detection volume $W(\vec{r})$ and the propagator for free diffusion P_{d} are described by exponentials (Gaussian functions). Therefore, spatial relationships can be factorized for each dimension xyz. As a result, Equation 12 can be written as a combination of transversal (2D) and longitudinal (3D) diffusion.

4.3 Autocorrelation of multiple species

Very often in FCS, one observes more than one dynamic property. Besides diffusion driven number fluctuations, a fluorophore usually shows some kind of inherent blinking, due to triplet state transitions (organic dyes) or protonation dependent quenching (GFPs) [43].

$$G(\tau) \overset{\text{def}}{=} G^{\text{T}}(\tau)G^{\text{D}}(\tau) = \left(1 + \frac{T}{1-T}\exp\left[-\frac{\tau}{\tau_{\text{trp}}}\right]\right) G^{\text{D}}(\tau) \tag{13}$$

Blinking increases the correlation amplitude $G(0)$ by the triplet fraction $1/(1-T)$. Accordingly, the average number of observed particles is decreased $n = (1-T)/G(0)$. In case of GFP blinking, two different blinking times have been described and the rate equations can get quite complicated. Besides photo-physics, the solution may contain mixtures of fluorescent particles with different dynamic properties, e.g. different mobility states or potential for transient binding. Such mixtures show several correlation times in the correlation curve. Equation 13 can be derived by considering the correlation functions of an ensemble, which can be built up by the contribution of n single molecules in the sample volume:

$$G(\tau) = \frac{g(\tau)}{\langle S(t)\rangle^2} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}(\tau)}{\langle S(t)\rangle^2} \tag{14}$$

with $g_{ij}(\tau)$ as the pairwise correlation function of identical ($i = j$) or distinguishable ($i \neq j$) particles

$$g_{ij}(\tau) = \langle s(t)\cdot s(t+\tau)\rangle = \frac{q_i q_j}{V}\int\int W(\vec{r})P_{\text{d},ij}\left(\vec{r}\,'|\vec{r},\tau\right)W(\vec{r}\,')dV\,dV' \tag{15}$$

Note that the diffusion propagator $P_{\text{d},ij}$ is now indexed, since the movement of some particle pairs may depend on each other and therefore show correlations. If particle i and particle j move independently, the mixed terms cancel $g_{ij}(\tau) = 0$. Due to the sums in Equation 14, adding up individual contributions of sub-ensembles is allowed. A frequently used expression to cover free diffusion of similarly labelled, differently sized particles is simply the sum of correlation functions, weighted with their relative fractions $F_k = nk/n$ to the overall amplitude $G(0) = 1/n$:

$$G^{\text{D}}(\tau) = \sum_{k=1}^{m} F_k G^{\text{D}}(\tau) = \frac{1}{n}\sum_{k=1}^{m} F_k \left(1 + \frac{\tau}{\tau_{\text{diff},k}}\right)^{-1}\left(1 + \frac{\tau}{SP^2\,\tau_{\text{diff},k}}\right) \tag{16}$$

Up to three diffusion times can usually be discriminated ($m = 3$) [17]. Note that this assumes homogenous molecular brightness of the different diffusion species. One of the molecular brightness values q_k is usually taken as a reference ($\alpha_k = q_k/q_1$). Brighter particles are over-represented [36]

$$G^{\text{D}}(\tau) = \frac{1}{n\left(\sum_k F_k\alpha_k\right)^2}\sum_k F_k\alpha_k^2 G_k^D(\tau) \tag{17}$$

Inhomogeneity in molecular brightness affects both the total concentration of observed particles as well as the real molar fractions F_k^{cor} [36]

$$n = \frac{1}{G^{\text{D}}(0)}\frac{\sum_k F_k^{\text{cor}}\alpha_k^2}{\left(\sum_k F_k^{\text{cor}}\alpha_k\right)^2} \quad \text{with} \quad F_k^{\text{cor}} = \frac{F_k/\alpha_k^2}{\sum_k F_k/\alpha_k} \tag{18}$$

4.4 Correcting non-correlated background signal

In FCS, the total signal is composed of the fluorescence and the non-correlated background: $S = F + B$. Non-correlated background signal like shot noise of the detectors or stray light decreases the relative fluctuation amplitude and must be corrected to derive true particle concentrations [12, 36]. In *PyCorrFit*, the background value [kHz] can be manually set for each channel (B1, B2) (Figure 1). For autocorrelation measurements ($B1 = B2 = B$) the average number of observed particles is then

$$n = \frac{1}{G^{\mathrm{D}}(0)} \left(\frac{S-B}{S} \right)^2 = \frac{1}{(1-T)G(0)} \left(\frac{S-B}{S} \right)^2 . \tag{19}$$

For dual-channel applications with cross-correlation (next section) the amplitudes must be corrected by contributions from each channel [39]

$$G_{\times,\mathrm{cor}}(0) = G_{\times,\mathrm{meas}}(0) \left(\frac{S_1}{S_1 - B_1} \right) \left(\frac{S_2}{S_2 - B_2} \right) \tag{20}$$

4.5 Cross-correlation

Cross-correlation is an elegant way to measure molecular interactions. The principle is to implement a dual-channel setup (e.g. channels 1 and 2), where two, interacting populations of molecules can be discriminated [7, 27, 33, 40]. In a dual-channel setup, complexes containing particles with both properties will evoke simultaneous signals in both channels. Such coincidence events can be extracted by cross-correlation between the two channels. A prominent implementation is dual-colour fluorescence cross-correlation spectroscopy (dcFCCS), where the binding partners are discriminated by spectrally distinct (differently coloured) labels. The formalism is similar to autocorrelation, just the origin of the signals must now be traced [31, 33, 40].

$$G_\times(\tau) \overset{\mathrm{def}}{=} G_{12}(\tau) = \frac{\langle \delta S_1(t) \delta S_2(t+\tau) \rangle}{\langle S_1(t) \rangle \langle S_2(t) \rangle} \approx \frac{\langle \delta S_2(t) \delta S_1(t+\tau) \rangle}{\langle S_1(t) \rangle \langle S_2(t) \rangle} = G_{21}(\tau) \tag{21}$$

A finite cross-correlation amplitude $G_{12}(0)$ indicates co-diffusion of complexes containing both types of interaction partners. The increase of the cross-correlation amplitude is linear for heterotypic binding but non-linear for homotypic interactions or higher order oligomers. The absolute magnitude of the cross-correlation amplitude must be calibrated because the chromatic mismatch of the detection volumes (different wavelength, different size) and their spatial displacement (d_x, d_y, d_z) constitute instrumental limits, even for 100% double labelled particles [40]. Equation 11 can be extended

$$G(\tau) = \frac{1}{n} \left(1 + \frac{4D_t\tau}{w_0^2} \right)^{-1} \left(1 + \frac{4D_t\tau}{z_0^2} \right)^{-1/2} \exp\left(- \frac{d_\mathrm{x}^2 + d_\mathrm{y}^2}{4D_t\tau + w_{0,\mathrm{eff}}} + \frac{d_\mathrm{z}^2}{4D_t\tau + z_{0,\mathrm{eff}}} \right) \tag{22}$$

The ratio between cross- and autocorrelation amplitude is used as a readout which can be linked to the degree of binding. Let us consider a heterodimerization, where channel 1 is sensitive for green labelled particles (g) and channel 2 is sensitive for red labelled particles (r), then the ratio of cross- and autocorrelation amplitudes is proportional to the fraction of ligand bound [40]

$$\begin{aligned} CC_1 &\overset{\mathrm{def}}{=} \frac{G_\times(0)}{G_1(0)} \propto \frac{c_{gr}}{c_r} \\ CC_2 &\overset{\mathrm{def}}{=} \frac{G_\times(0)}{G_2(0)} \propto \frac{c_{gr}}{c_g} \end{aligned} \tag{23}$$

Recently, a correction for bleed-through of the signals between the two channels has been worked out [3]. The effect on binding curves measured with cross-correlation can be quite dramatic [39]. To treat spectral cross-talk, the experimenter has to determine with single coloured probes how much of the signal (ratio in %) is detected by the orthogonal, 'wrong' channel (BT_{12}, BT_{12}). Usually the bleed-through from the red into the green channel can be neglected ($BT_{21} = 0$) leaving only a contribution of bleed through from the green into the red channel (BT_{12})

$$\begin{aligned} \langle F_1 \rangle &= \langle \hat{F}_1 \rangle + \langle \hat{F}_2 \rangle BT_{21} \cong \langle \hat{F}_1 \rangle \\ \langle F_2 \rangle &= \langle \hat{F}_2 \rangle + \langle \hat{F}_1 \rangle BT_{12} \end{aligned} \tag{24}$$

Here, the dashed fluorescence signals are the true contributions from single labelled species. Thus, each set of simultaneously recorded auto and cross-correlation curves suffers from a specific fraction of wrong signal in the vulnerable red channel, $X_2 = BT_{12}\langle \hat{F}_1 \rangle / \hat{F}_2 \rangle$, which can be used to back-correct the cross-correlation amplitudes [3, 39]

$$\frac{c_{gr}}{c_r} \propto \frac{CC_1 - X_2}{(1 - X_2)} \tag{25a}$$

$$\frac{c_{gr}}{c_g} \propto \frac{CC_2 - X_2\,(1 - X_2)\,\frac{G_1^{\mathrm{D}}(0)}{G_2^{\mathrm{D}}(0)}}{1 + X_2 \frac{G_1^{\mathrm{D}}(0)}{G_2^{\mathrm{D}}(0)} - 2X_2 CC_2} \tag{25b}$$

As apparent from Equations 25, it is much simpler to use the autocorrelation amplitude measured in the green channel for normalization (25a) and not the cross-talk affected red channel (25b). Finally, the proportionality between the fraction ligand bound and the measured cross-correlation ratio depend solely on the effective detection volumes of all three channels (two auto- and the cross-correlation channels) and must be determined with appropriate positive controls (single labelled and double labelled calibration dyes).

4.6 Extensions of the method

Using as confocal alignment for FCS measurements is very successful and widely applied. However, mainly in the context of membrane research, other excitation schemes have been explored. Once the excitation and detection geometry is changed one has to account for it in the model functions. This is usually done by modifying the expressions of the instrumental detection efficiency $W(\vec{r})$ in Equation 10. However, in many cases analytical solutions to the above integrals are not straightforward and approximations have to be made. The following section introduces model functions for different detection symmetries and particle dynamics.

4.6.1 Perpendicular scanning FCS (pSFCS)

Scanning FCS with a scan path perpendicular to the membrane plane was introduced to measure the lateral diffusion of fluorescent components of a fluid lipid membrane [28]. Using the linear scan option of a confocal laser scanning microscope (CLSM), the focal spot is moved repeatedly on a straight line through a free standing membrane (typically the equatorial face of a giant vesicle). The fluorescence photons emitted from the detection volume are continuously recorded and stored. The signal originating from the transit through the membrane must be extracted post-measurement, correlated, and evaluated by well-known FCS model functions. Scanning can also be performed with two parallel scan paths perpendicular through the membrane (continuous imaging with two lines). Like in two-focus

FCS [5], the distance between these scan paths can been used to calibrate the diffusion coefficients without further use of diffusion standards. We recently introduced a free, open source software tool (*PyScanFCS*) to perform these steps [18].

4.6.2 Circular scanning FCS (cSFCS)

The principle of circular scanning FCS is similar. Here, the laser focus is moved in small circles, several μm in diameter. While pSFCS requires relatively large free standing membranes, cSFCS can be performed in μm sized homogeneous fluorescent regions of supported bilayers, cell membranes, as well as the apical poles of giant vesicles. Once the radius of the circular scan path has been accurately determined (e.g. with a fluorescent grid), the diffusion coefficients can be determined from cross-correlation curves without the need to calibrate the detection volume by diffusion standards [22, 23] (See Figure 3).

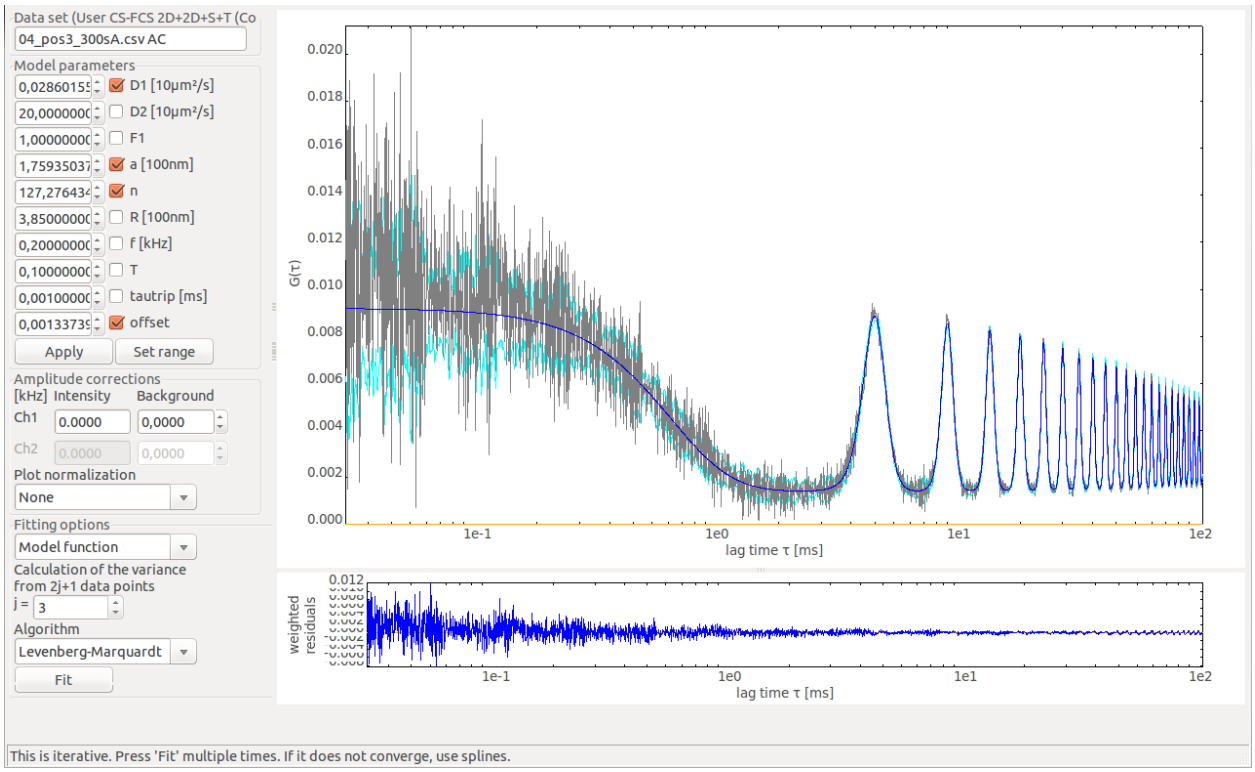

Figure 3, user interface with external model function: cSFCS curve of DiO diffusing in a reconstituted lipid bilayer on glass support. Fitting yields a diffusion coefficient of $0.28\,\mu m^2 s^{-1}$ ($F1 = 1$, so only one component is fitted). The source code of the external model function for this fit is shown in Figure 2.

4.6.3 Total internal reflection FCS (TIR-FCS)

TIR-FCS was developed to measure transient ligand binding events to receptors in the membrane [35]. In contrast to scanning FCS, the detection volume is fixed. However, the geometry is confined to the surface showing an exponential decay of the evanescent field along z, whereas the lateral boundaries imposed in xy by a pinhole in the detection path. The situation is notoriously difficult to model and different solutions have been proposed.

TIR-FCS with Gaussian-shaped lateral detection volume. The detection volume is axially confined by an evanescent field and has an effective size of

$$V = \pi R_0^2 d_{\text{eva}} \tag{26}$$

where R_0 is the lateral extent of the detection volume and d_{eva} is the evanescent field depth[10]. From the concentration C, the effective number of particles is $n = CV$. The decay constant κ is the inverse of the depth d_{eva} :

$$d_{\text{eva}} = \frac{1}{\kappa} \tag{27}$$

TIR-FCS with a square-shaped lateral detection volume. The detection volume is axially confined by an evanescent field of depth $d_{\text{eva}} = 1/\kappa$. The lateral detection area is a convolution of the point spread function of the microscope of size σ,

$$\sigma = \sigma_0 \frac{\lambda}{NA}, \tag{28}$$

with a square of side length a.

4.7 Fitting

One can define a distance $d(G, H)$ between two discrete functions G and H with the discrete domain of definition $\tau_1 \dots \tau_n$ as the sum of squares:

$$d(G, H) = \sum_{i=1}^{n} \left[G(\tau_i) - H(\tau_i)\right]^2 \tag{29}$$

The least-squares method minimizes this distance between the model function G and the experimental values H by modifying k additional fitting parameters $\alpha_1, \dots, \alpha_k$:

$$\chi^2 = \min_{\alpha_1,\dots,\alpha_k} \sum_{i=1}^{n} \left[G(\tau_i, \alpha_1, \dots, \alpha_k) - H(\tau_i)\right]^2 \tag{30}$$

The minimum distance χ^2 is used to characterize the success of a fit. Note, that if the number of fitting parameters k becomes too large, multiple values for χ^2 can be found, depending on the starting values of the k parameters.

4.7.1 Weighted fitting

In certain cases, it is useful to implement weights (standard deviation) σ_i for the calculation of χ^2. For example, very noisy parts of a correlation curve can falsify the resulting fit. In *PyCorrFit*, weighting is implemented as follows:

$$\chi^2_{\text{weighted}} = \min_{\alpha_1,\dots,\alpha_k} \sum_{i=1}^{n} \frac{\left[G(\tau_i, \alpha_1, \dots, \alpha_k) - H(\tau_i)\right]^2}{\sigma_i^2} \tag{31}$$

PyCorrFit is able to calculate the weights σ_i from the experimental data. The different approaches of this calculation of weights implemented in *PyCorrFit* are explained in Section 1.4.

[10] Where the field has decayed to $1/e$

4.7.2 Algorithms

PyCorrFit uses the non-linear least-squares fitting capabilities from `scipy.optimize`. This package contains several algorithms to minimize the sum of the squares. PyCorrFit can utilize several algorithms to perform this minimization. The descriptions of the algorithms listed here are partly copied from the scipy documentation at `http://docs.scipy.org/doc/scipy/reference/optimize.html`.

- The **BFGS** method uses the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS) [10] (pp. 136). It uses the first derivatives only. BFGS has proven good performance even for non-smooth optimizations.
- The **Levenberg-Marquardt** algorithm [14] uses the first derivatives and combines the Gauss–Newton algorithm with a trust region approach. It is very robust compared to other algorithms and it is very popular in curve-fitting. *PyCorrFit* uses this algorithm by default. If this algorithm is used, *PyCorrFit* can estimate an error of the fit parameters using the covariance matrix.
- The **Nelder-Mead** method uses the Simplex algorithm [19, 45]. This algorithm has been successful in many applications but other algorithms using the first and/or second derivatives information might be preferred for their better performances and robustness in general.
- The method **Polak-Ribiere** uses a nonlinear conjugate gradient algorithm by Polak and Ribiere, a variant of the Fletcher-Reeves method described in [10] pp. 120-122. Only the first derivatives are used.
- The method **Powell** is a modification of Powell's method [24, 25] which is a conjugate direction method. It performs sequential one-dimensional minimizations along each vector of the directions set, which is updated at each iteration of the main minimization loop. The function need not be differentiable, and no derivatives are taken.

5 Implemented model functions

This is an overview of all the model functions that are implemented in *PyCorrFit*. To each model a unique model ID is assigned. Most of the following information is also accessible from within *PyCorrFit* using the *Page info* tool.

5.1 Confocal FCS

Name	**Confocal (Gaussian) T+3D**
ID	**6011**
Descr.	Three-dimensional free diffusion with a Gaussian laser profile (eliptical), including a triplet component [9, 43, 44].

$$G(\tau) = A_0 + \frac{1}{n}\frac{1}{(1+\tau/\tau_{\mathrm{diff}})}\frac{1}{\sqrt{1+\tau/(SP^2\tau_{\mathrm{diff}})}}\left(1+\frac{Te^{-\tau/\tau_{\mathrm{trip}}}}{1-T}\right) \tag{32}$$

A_0	Offset
n	Effective number of particles in confocal volume
τ_{diff}	Characteristic residence time in confocal volume
SP	Structural parameter, describes elongation of the confocal volume
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet

Name **Confocal (Gaussian) T+3D+3D**
ID **6030**
Descr. Two-component three-dimensional free diffusion with a Gaussian laser profile, including a triplet component [1, 6, 21].

$$G(\tau) = A_0 + \frac{1}{n(F+\alpha(1-F))^2}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\times \\ \times\left[\frac{F}{(1+\tau/\tau_1)}\frac{1}{\sqrt{1+\tau/(SP^2\tau_1)}}+\alpha^2\frac{1-F}{(1+\tau/\tau_2)}\frac{1}{\sqrt{1+\tau/(SP^2\tau_2)}}\right] \tag{33}$$

A_0	Offset
n	Effective number of particles in confocal volume ($n = n_1 + n_2$)
τ_1	Diffusion time of particle species 1
τ_2	Diffusion time of particle species 2
F	Fraction of molecules of species 1 ($n_1 = Fn$)
α	Relative molecular brightness of particles 1 and 2 ($\alpha = q_2/q_1$)
SP	Structural parameter, describes elongation of the confocal volume
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

Name **Confocal (Gaussian) T+2D**
ID **6002**
Descr. Two-dimensional diffusion with a Gaussian laser profile, including a triplet component [1, 9, 26, 30, 43, 44].

$$G(\tau) = A_0 + \frac{1}{n}\frac{1}{(1+\tau/\tau_{\text{diff}})}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{34}$$

A_0	Offset
n	Effective number of particles in confocal area
τ_{diff}	Characteristic residence time in confocal area
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

Name **Confocal (Gaussian) T+2D+2D**
ID **6031**
Descr. Two-component, two-dimensional diffusion with a Gaussian laser profile, including a triplet component [1, 6, 21].

$$G(\tau) = A_0 + \frac{1}{n(F+\alpha(1-F))^2}\left[\frac{F}{1+\tau/\tau_1}+\alpha^2\frac{1-F}{1+\tau/\tau_2}\right]\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{35}$$

A_0	Offset
n	Effective number of particles in confocal area ($n = n_1 + n_2$)
τ_1	Diffusion time of particle species 1
τ_2	Diffusion time of particle species 2
F	Fraction of molecules of species 1 ($n_1 = Fn$)
α	Relative molecular brightness of particles 1 and 2 ($\alpha = q_2/q_1$)
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

Name **Confocal (Gaussian) T+3D+2D**
ID **6032**
Descr. Two-component, two- and three-dimensional diffusion with a Gaussian laser profile, including a triplet component [1, 6, 21].

$$G(\tau) = A_0 + \frac{1}{n(1-F+\alpha F)^2}\left[\frac{1-F}{1+\tau/\tau_{2D}} + \frac{\alpha^2 F}{(1+\tau/\tau_{3D})}\frac{1}{\sqrt{1+\tau/(SP^2\tau_{3D})}}\right]\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{36}$$

A_0	Offset
n	Effective number of particles in confocal volume ($n = n_{2D} + n_{3D}$)
τ_{2D}	Diffusion time of surface bound particles
τ_{3D}	Diffusion time of freely diffusing particles
F	Fraction of molecules of the freely diffusing species ($n_{3D} = Fn$)
α	Relative molecular brightness of particle species ($\alpha = q_{3D}/q_{2D}$)
SP	Structural parameter, describes elongation of the confocal volume
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

5.2 TIR-FCS

The model functions make use of the Faddeeva function (complex error function)[11]:

$$\begin{aligned} w(i\xi) &= e^{\xi^2}\operatorname{erfc}(\xi) \\ &= e^{\xi^2}\cdot\frac{2}{\sqrt{\pi}}\int_{\xi}^{\infty}\mathrm{e}^{-\alpha^2}\mathrm{d}\alpha \end{aligned} \tag{37}$$

The lateral detection area has the same shape as in confocal FCS. Thus, correlation functions for two-dimensional diffusion of the confocal case apply and are not mentioned here.

[11] In user-defined model functions (Section 3.1), the Faddeeva function is accessible through `wofz()`. For convenience, the function `wixi()` can be used which only takes ξ as an argument and the imaginary i can be omitted.

5.2.1 TIR-FCS with Gaussian-shaped lateral detection volume

Name **TIR (Gaussian/Exp.) T+3D**
ID **6014**
Descr. Three-dimensional free diffusion with a Gaussian lateral detection profile and an exponentially decaying profile in axial direction, including a triplet component [8, 20, 34].

$$G(\tau) = \frac{1}{C}\frac{\kappa^2}{\pi(R_0^2+4D\tau)}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\left(\sqrt{\frac{D\tau}{\pi}}+\frac{1-2D\tau\kappa^2}{2\kappa}w\left(i\sqrt{D\tau}\kappa\right)\right) \tag{38}$$

C	Particle concentration in confocal volume
κ	Evanescent decay constant ($\kappa = 1/d_{\text{eva}}$)
R_0	Lateral extent of the detection volume
D	Diffusion coefficient
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

Name **TIR (Gaussian/Exp.) T+3D+3D**
ID **6034**
Descr. Two-component three-dimensional diffusion with a Gaussian lateral detection profile and an exponentially decaying profile in axial direction, including a triplet component [8, 20, 34].

$$\begin{aligned} G(\tau) =& A_0 + \frac{1}{n(1-F+\alpha F)^2}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\times \\ &\times\left[\frac{F\kappa}{1+4D_1\tau/R_0^2}\left(\sqrt{\frac{D_1\tau}{\pi}}+\frac{1-2D_1\tau\kappa^2}{2\kappa}w\left(i\sqrt{D_1\tau}\kappa\right)\right)+\right. \\ &\left.+\frac{(1-F)\alpha^2\kappa}{1+4D_2\tau/R_0^2}\left(\sqrt{\frac{D_2\tau}{\pi}}+\frac{1-2D_2\tau\kappa^2}{2\kappa}w\left(i\sqrt{D_2\tau}\kappa\right)\right)\right] \end{aligned} \tag{39}$$

A_0	Offset
n	Effective number of particles in confocal volume ($n = n_1 + n_2$)
D_1	Diffusion coefficient of species 1
D_2	Diffusion coefficient of species 2
F	Fraction of molecules of species 1 ($n_1 = Fn$)
α	Relative molecular brightness of particle species ($\alpha = q_2/q_1$)
R_0	Lateral extent of the detection volume
κ	Evanescent decay constant ($\kappa = 1/d_{\text{eva}}$)
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

Name **TIR (Gaussian/Exp.) T+3D+2D**
ID **6033**
Descr. Two-component, two- and three-dimensional diffusion with a Gaussian lateral detection profile and an exponentially decaying profile in axial direction, including a triplet component [8,20,34].

$$G(\tau) = A_0 + \frac{1}{n(1-F+\alpha F)^2}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\times \tag{40}$$
$$\times\left[\frac{1-F}{1+4D_{2\text{D}}\tau/R_0^2}+\frac{\alpha^2 F\kappa}{1+4D_{3\text{D}}\tau/R_0^2}\left(\sqrt{\frac{D_{3\text{D}}\tau}{\pi}}+\frac{1-2D_{3\text{D}}\tau\kappa^2}{2\kappa}w\left(i\sqrt{D_{3\text{D}}\tau}\kappa\right)\right)\right]$$

A_0	Offset
n	Effective number of particles in confocal volume ($n = n_{2\text{D}} + n_{3\text{D}}$)
$D_{2\text{D}}$	Diffusion coefficient of surface bound particles
$D_{3\text{D}}$	Diffusion coefficient of freely diffusing particles
F	Fraction of molecules of the freely diffusing species ($n_{3\text{D}} = Fn$)
α	Relative molecular brightness of particle species ($\alpha = q_{3\text{D}}/q_{2\text{D}}$)
R_0	Lateral extent of the detection volume
κ	Evanescent decay constant ($\kappa = 1/d_{\text{eva}}$)
T	Fraction of particles in triplet (non-fluorescent) state
τ_{trip}	Characteristic residence time in triplet state

5.2.2 TIR-FCS with a square-shaped lateral detection volume

Name **TIR (□xσ/Exp.) 3D**
ID **6010**
Descr. Three-dimensional diffusion with a square-shaped lateral detection area taking into account the size of the point spread function; and an exponential decaying profile in axial direction [29,46].

$$G(\tau) = \frac{\kappa^2}{C}\left(\sqrt{\frac{D\tau}{\pi}}+\frac{1-2D\tau\kappa^2)}{2\kappa}w\left(i\sqrt{D\tau}\kappa\right)\right)\times \tag{41}$$
$$\times\left[\frac{2\sqrt{\sigma^2+D\tau}}{\sqrt{\pi}a^2}\left(\exp\left(-\frac{a^2}{4(\sigma^2+D\tau)}\right)-1\right)+\frac{1}{a}\operatorname{erf}\left(\frac{a}{2\sqrt{\sigma^2+D\tau}}\right)\right]^2$$

C	Particle concentration in detection volume
σ	Lateral size of the point spread function
a	Side size of the square-shaped detection area
κ	Evanescent decay constant ($\kappa = 1/d_{\text{eva}}$)
D	Diffusion coefficient

Name **TIR (□xσ/Exp.) 3D+3D**
ID **6023**
Descr. Two-component three-dimensional free diffusion with a square-shaped lateral detection area taking into account the size of the point spread function; and an exponential decaying profile in axial direction.
The correlation function is a superposition of three-dimensional model functions of the type **3D (□xσ)** (6010) [29, 46].

Name **TIR (□xσ) 2D**
ID **6000**
Descr. Two-dimensional diffusion with a square-shaped lateral detection area taking into account the size of the point spread function [29, 46][12].

$$G(\tau) = \frac{1}{C}\left[\frac{2\sqrt{\sigma^2 + D\tau}}{\sqrt{\pi}a^2}\left(\exp\left(-\frac{a^2}{4(\sigma^2 + D\tau)}\right) - 1\right) + \frac{1}{a}\operatorname{erf}\left(\frac{a}{2\sqrt{\sigma^2 + D\tau}}\right)\right]^2 \tag{42}$$

C Particle concentration in detection area
σ Lateral size of the point spread function
a Side size of the square-shaped detection area
D Diffusion coefficient

Name **TIR (□xσ) 2D+2D**
ID **6022**
Descr. Two-component two-dimensional diffusion with a square-shaped lateral detection area taking into account the size of the point spread function.
The correlation function is a superposition of two-dimensional model functions of the type **2D (□xσ)** (6000) [29, 46].

Name **TIR (□xσ/Exp.) 3D+2D**
ID **6020**
Descr. Two-component two- and three-dimensional diffusion with a square-shaped lateral detection area taking into account the size of the point spread function; and an exponential decaying profile in axial direction.
The correlation function is a superposition of the two-dimensional model function **2D (□xσ)** (6000) and the three-dimensional model function **3D (□xσ)** (6010) [29, 46].

Name	**TIR (□xσ/Exp.) 3D+2D+kin**
ID	**6021**
Descr.	Two-component two- and three-dimensional diffusion with a square-shaped lateral detection area taking into account the size of the point spread function; and an exponential decaying profile in axial direction. This model covers binding and unbinding kintetics. The correlation function for this model was introduced in [29]. Because approximations are made in the derivation, please verify if this model is applicable to your problem before using it.

6 Troubleshooting

If you are having problems with PyCorrFit, you might find the solution in the frequently asked questions[13] or on other pages in the *PyCorrFit* wiki[14]. There you will also find instructions on how to contact us to file a bug or to request a feature.

[13] https://github.com/paulmueller/PyCorrFit/wiki/Frequently-Asked-Questions-%28FAQ%29
[14] https://github.com/paulmueller/PyCorrFit/wiki

Acknowledgements

I thank André Scholich (TU Dresden, Germany) for initial proof reading of the manuscript.

References

[1] S. R. Aragon and R. Pecora. Fluorescence correlation spectroscopy as a probe of molecular dynamics. *The Journal of Chemical Physics*, 64(4):1791–1803, 1976. doi:10.1063/1.432357.

[2] K. Bacia, S. A. Kim, and P. Schwille. Fluorescence cross-correlation spectroscopy in living cells. *Nat Methods*, 3(2):83–9, 2006. doi:10.1038/nmeth822.

[3] K. Bacia, Zdeněk Petrášek, and P. Schwille. Correcting for spectral cross-talk in dual-color fluorescence cross-correlation spectroscopy. *Chemphyschem*, 13(5):1221–31, 2012. doi:10.1002/cphc.201100801.

[4] M. Brinkmeier, K. Dörre, J. Stephan, and M. Eigen. Two-beam cross-correlation: a method to characterize transport phenomena in micrometer-sized structures. *Analytical Chemistry*, 71(3):609–616, Feb 1999. doi:10.1021/ac980820i.

[5] Thomas Dertinger, Victor Pacheco, Iris von der Hocht, Rudolf Hartmann, Ingo Gregor, and Jörg Enderlein. Two-focus fluorescence correlation spectroscopy: A new tool for accurate and absolute diffusion measurements. *ChemPhysChem*, 8(3):433–443, 2007. doi:10.1002/cphc.200600638.

[6] Elliot L. Elson and Douglas Magde. Fluorescence correlation spectroscopy. i. conceptual basis and theory. *Biopolymers*, 13(1):1–27, 1974. doi:10.1002/bip.1974.360130102.

[7] Y. H. Foo, N. Naredi-Rainer, D. C. Lamb, S. Ahmed, and T. Wohland. Factors affecting the quantification of biomolecular interactions by fluorescence cross-correlation spectroscopy. *Biophys J*, 102(5):1174–83, 2012. doi:10.1016/j.bpj.2012.01.040.

[8] Kai Hassler, Tiemo Anhut, Rudolf Rigler, Michael Gösch, and Theo Lasser. High count rates with total internal reflection fluorescence correlation spectroscopy. *Biophysical Journal*, 88(1):L01–L03, January 2005. doi:10.1529/biophysj.104.053884.

[9] Ulrich Haupts, Sudipta Maiti, Petra Schwille, and Watt W. Webb. Dynamics of fluorescence fluctuations in green fluorescent protein observed by fluorescence correlation spectroscopy. *Proceedings of the National Academy of Sciences*, 95(23):13573–13578, 1998. doi:10.1073/pnas.95.23.13573.

[10] Nocedal J. and Wright S J. *Numerical Optimization*. Springer Berlin Heidelberg, 2006. doi:10.1007/978-3-540-35447-5.

[11] S. A. Kim, K. G. Heinze, and P. Schwille. Fluorescence correlation spectroscopy in living cells. *Nat Methods*, 4(11):963–73, 2007. doi:10.1038/nmeth1104.

[12] D. Koppel. Statistical accuracy in fluorescence correlation spectroscopy. *Phys Rev A*, 10:1938–1945, 1974. doi:10.1103/physreva.10.1938.

[13] J. Korlach, P. Schwille, W. W. Webb, and G. W. Feigenson. Characterization of lipid bilayer phases by confocal microscopy and fluorescence correlation spectroscopy. *Proc Natl Acad Sci U S A*, 96(15):8461–6, 1999. doi:10.1073/pnas.96.15.8461.

[14] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[15] D. Magde, W. W. Webb, and E. L. Elson. Fluorescence correlation spectroscopy. iii. uniform translation and laminar flow. *Biopolymers*, 17:361–376, 1978. doi:10.1002/bip.1978.360170208.

[16] Douglas Magde, Elliot L. Elson, and Watt W. Webb. Fluorescence correlation spectroscopy. ii. an experimental realization. *Biopolymers*, 13(1):29–61, 1974. doi:10.1002/bip.1974.360130103.

[17] U. Meseth, T. Wohland, R. Rigler, and H. Vogel. Resolution of fluorescence correlation measurements. *Biophys J*, 76:1619–1631, 1999. doi:10.1016/S0006-3495(99)77321-2.

[18] P. Müller, P. Schwille, and T. Weidemann. Scanning fluorescence correlation spectroscopy (SFCS) with a scan path perpendicular to the membrane plane. *Methods Mol Biol*, 1076:635–51, 2014. doi:10.1007/978-1-62703-649-8_29.

[19] John A Nelder and Roger Mead. A simplex method for function minimization. *Computer journal*, 7(4):308–313, 1965. doi:10.1093/comjnl/7.4.308.

[20] Yu Ohsugi, Kenta Saito, Mamoru Tamura, and Masataka Kinjo. Lateral mobility of membrane-binding proteins in living cells measured by total internal reflection fluorescence correlation spectroscopy. *Biophysical Journal*, 91(9):3456–3464, 2006. doi:10.1529/biophysj.105.074625.

[21] A. G. Palmer and N. L. Thompson. Theory of sample translation in fluorescence correlation spectroscopy. *Biophysical Journal*, 51(2):339–343, Feb 1987. doi:10.1016/S0006-3495(87)83340-4.

[22] Zdeněk Petrášek, J. Ries, and P. Schwille. Scanning FCS for the characterization of protein dynamics in live cells. *Methods Enzymol*, 472:317–43, 2010. doi:10.1016/s0076-6879(10)72005-x.

[23] Zdeněk Petrášek and Petra Schwille. Precise measurement of diffusion coefficients using scanning fluorescence correlation spectroscopy. *Biophysical Journal*, 94(4):1437–1448, February 2008. doi:10.1529/biophysj.107.108811.

[24] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, Feb 1964. doi:10.1093/comjnl/7.2.155.

[25] William Press, Brian P Flannery, SAUL Teukolsky, and WT Vetterling. Numerical recipes. *Cambridge University Press*, 1:989, 2006.

[26] Hong Qian and Elliot L. Elson. Analysis of confocal laser-microscope optics for 3-d fluorescence correlation spectroscopy. *Applied Optics*, 30(10):1185–1195, Apr 1991. doi:10.1364/AO.30.001185.

[27] J. Ries, Zdeněk Petrášek, A. J. Garcia-Saez, and P. Schwille. A comprehensive framework for fluorescence cross-correlation spectroscopy. *New Journal of Physics*, 12(11):113009, 2010. doi:10.1088/1367-2630/12/11/113009.

[28] J. Ries and P. Schwille. Studying slow membrane dynamics with continuous wave scanning fluorescence correlation spectroscopy. *Biophys J*, 91(5):1915–24, 2006. doi:10.1529/biophysj.106.082297.

[29] Jonas Ries and Petra Schwille. New concepts for fluorescence correlation spectroscopy on membranes. *Physical Chemistry Chemical Physics*, 10(24):–, 2008. doi:10.1039/b718132a.

[30] R. Rigler, Ü. Mets, J. Widengren, and P. Kask. Fluorescence correlation spectroscopy with high count rate and low background: analysis of translational diffusion. *European Biophysics Journal*, 22:169–175, 1993. doi:10.1007/BF00185777.

[31] K. Rippe. Simultaneous binding of two DNA duplexes to the ntrc-enhancer complex studied by two-color fluorescence cross-correlation spectroscopy. *Biochemistry*, 39(9):2131–9, 2000. doi:10.1021/bi9922190.

[32] P. Schwille, U. Haupts, S. Maiti, and W. W. Webb. Molecular dynamics in living cells observed by fluorescence correlation spectroscopy with one- and two-photon excitation. *Biophys J*, 77(4):2251–65, 1999. doi:10.1016/s0006-3495(99)77065-7.

[33] P. Schwille, F.J. Meyer-Almes, and R. Rigler. Dual-color fluorescence cross-correlation spectroscopy for multicomponent diffusional analysis in solution. *Biophysical Journal*, 72(4):1878–1886, April 1997. doi:10.1016/s0006-3495(97)78833-7.

[34] Tammy E. Starr and Nancy L. Thompson. Total internal reflection with fluorescence correlation spectroscopy: Combined surface reaction and solution diffusion. *Biophysical Journal*, 80(3):1575 – 1584, 2001. doi:10.1016/S0006-3495(01)76130-9.

[35] N. L. Thompson and B. L. Steele. Total internal reflection with fluorescence correlation spectroscopy. *Nat Protoc*, 2(4):878–90, 2007. doi:10.1038/nprot.2007.110.

[36] N.L. Thompson. *Fluorescence Correlation Spectroscopy*, volume 1 of *Topics in Fluorescence Spectroscopy*, pages 337–378. Plenum Press, New York, techniques edition, 1991.

[37] M. Wachsmuth, W. Waldeck, and J. Langowski. Anomalous diffusion of fluorescent probes inside living cell nuclei investigated by spatially-resolved fluorescence correlation spectroscopy. *J Mol Biol*, 298(4):677–89, 2000. doi:10.1006/jmbi.2000.3692.

[38] T. Weidemann and P. Schwille. *Fluorescence Correlation Spectroscopy in Living Cells*. Handbook of Single-Molecule Biophysics. Springer, Heidelberg, 2009.

[39] T. Weidemann and P. Schwille. Dual-color fluorescence cross-correlation spectroscopy with continuous laser excitation in a confocal setup. *Methods Enzymol*, 518:43–70, 2013. doi:10.1016/B978-0-12-388422-0.00003-0.

[40] T. Weidemann, M. Wachsmuth, M Tewes, K. Rippe, and J. Langowski. Analysis of ligand binding by two-colour fluorescence cross-correlation spectroscopy. *Single Mol*, 3(1):49–61, 2002. doi:10.1002/1438-5171(200204)3:1<49::aid-simo49>3.0.co;2-t.

[41] M. Weiss, H. Hashimoto, and T. Nilsson. Anomalous protein diffusion in living cells as seen by fluorescence correlation spectroscopy. *Biophys J*, 84(6):4043–4052, 2003. doi:10.1016/s0006-3495(03)75130-3.

[42] J. Widengren and R. Rigler. Fluorescence correlation spectroscopy as a tool to investigate chemical reactions in solutions and on cell surfaces. *Cell Mol Biol (Noisy-le-grand)*, 44(5):857–79, 1998. Retrieved from http://europepmc.org/abstract/MED/9764752.

[43] Jerker Widengren, Ülo Mets, and Rudolf Rigler. Fluorescence correlation spectroscopy of triplet states in solution: a theoretical and experimental study. *The Journal of Physical Chemistry*, 99(36):13368–13379, 1995. doi:10.1021/j100036a009.

[44] Jerker Widengren, Rudolf Rigler, and Ülo Mets. Triplet-state monitoring by fluorescence correlation spectroscopy. *Journal of Fluorescence*, 4:255–258, 1994. doi:10.1007/BF01878460.

[45] M.H. Wright. Direct search methods: Once scorned, now respectable. In D.F. Griffiths and G.A. Watson, editors, *Numerical Analysis*, pages 191–208. Addison Wesley Longman, Harlow, UK, 1996.

[46] Stoyan Yordanov, Andreas Best, Klaus Weisshart, and Kaloian Koynov. Note: An easy way to enable total internal reflection-fluorescence correlation spectroscopy (TIR-FCS) by combining commercial devices for FCS and TIR microscopy. *Review of Scientific Instruments*, 82(3):036105, 2011. doi:10.1063/1.3557412.