PyCorrFit - Generic cross-platform FCS fitting tool
Software Guide

Paul Müller
Biotechnology Center of the TU Dresden
October 28, 2013

Contents

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. The program source code is available at GitHub[2]. Please do not hesitate to sign up and add a feature request. If you you found a bug, please let me 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.

What PyCorrFit can do

- Load correlation curves from numerous correlators
- Process these curves (e.g. background correction, s. Tools section 3.1.1)
- 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

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. PyCorrFit 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`.
2. APD: Photon Counting Device from PerkinElmer Optoelectronics
   Correlator: ALV-6000
3. LSM Confocor2 or Confocor3 setups from Zeiss, Germany.

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

1.2.2 Software

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

- **MacOSx**. Binary files for MacOSx >10.6.8 are available from the download page but have not yet been fully tested for stability.
- **Windows**. For Windows XP or Windows 7, stand-alone binary executables are available from the download page.
- **Linux**. There are executable binaries for widely used distributions (e.g. Ubuntu).
- **Sources** 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. To obtain the latest source, visit PyCorrFit at GitHub (https://github.com/paulmueller/PyCorrFit). PyCorrFit depends on the following python modules:

  ```
  python-matplotlib (≥ 1.0.1)
  python-numpy (≥ 1.5.1)
  python-scipy (≥ 0.8.0)
  python-sympy (≥ 0.7.2)
  python-yaml
  python-wxtools
  python-wxgtk2.8-dbg
  ```

  For older versions of Ubuntu, some of the above package versions are not listed in the package repository. To enable the use of PyCorrFit on those systems, the following tasks have to be performed:

  matplotlib. The tukss-ppa includes version 1.0.1. After adding the repository (`apt-add-repository ppa:tukss/ppa`), matplotlib can be installed as usual.

  numpy. The package from a later version of Ubuntu can be installed: https://launchpad.net/ubuntu/+source/python-numpy/

  scipy. The package from a later version of Ubuntu can be installed: https://launchpad.net/ubuntu/+source/python-scipy/

  sympy. To enable importing external model functions, sympy is required. It is available from http://code.google.com/p/sympy/downloads/list. Unpacking the archive and executing `python setup.py install` within the unpacked directory will install sympy.

Alternatively `python-pip` (http://pypi.python.org/pypi/pip) can be used to install up-to-date python modules.
LaTeX. PyCorrFit can save correlation curves as images using matplotlib. It is also possible to utilize Latex to 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, the packages LaTeX, dvipng, Ghostscript and the scientific latex packages `texlive-science` and `texlive-math-extra` need to be installed.

2 Theoretical background

2.1 Derivation of FCS model functions

This section introduces the calculation of FCS model functions. It supplies some background information and points out general properties of correlation functions.

2.1.1 General Autocorrelation function for a single species

FCS model functions describe how the signal $F(t)$, emitted from a certain observation volume, is temporally dependent on its own past (autocorrelation) or on some other signal (cross-correlation). The autocorrelation $G(\tau)$ of a signal $F(t)$ is computed as follows:

$$G(\tau) = \frac{\langle \delta F(t) \delta F(t+\tau) \rangle}{\langle F(t) \rangle^2} = \frac{g(\tau)}{\langle F(t) \rangle^2}. \tag{1}$$

$G(\tau)$ normalized autocorrelation curve

τ lag time

$\langle F \rangle$ the expectation value of $F(t)$. Applying the ergodic theorem, this can be rewritten as the time average

$$\langle F(t) \rangle = \lim_{T \to \text{inf}} \frac{1}{T} \int_0^T F(t) \mathrm{d}t.$$

$\delta F(t)$ $= F(t) - \langle F(t) \rangle$ fluctuation of the fluorescence signal

$g(\tau)$ non normalized autocorrelation curve

The fluorescence signal is dependent on the size and shape of the detection volume (e.g. Gaussian shaped for confocal setups or exponential decaying for TIRF setups), on the propagator of the diffusing dye (free diffusion, diffusion with flow, etc.), and the brightness and concentration of the dye under observation [2].

$$G(\tau) = \frac{q^2 C \int \mathrm{d}^3 r \int \mathrm{d}^3 r' \, \Omega(\mathbf{r}) \Phi(\mathbf{r}, \mathbf{r}', \tau) \Omega(\mathbf{r}')}{\langle F(t) \rangle^2} \tag{2}$$

q molecular brightness, dependent on excitation intensity, quantum yield, i.e. emission properties and absorption cross sections of the dye, and the detection efficiency of the instrument.

Ω 3D molecule detection function, dependent on the shape of the pinholes used for detection and the excitation laser profile, i.e. the point spread function (PSF).

Φ diffusion propagator. The distribution of dyes in a liquid follows Fick's laws of diffusion. For free diffusion, this is a simple Gaussian distribution.

F fluorescence signal of the sample. It is defined as

$$F(t) = q \int \mathrm{d}^3 r \, \Omega(\mathbf{r}) c(\mathbf{r}, t)$$

with $c(\mathbf{r}, t)$ being dye distribution (particle concentration) inside the detection volume.

C average concentration of the dye following the dynamics of the propagator Φ. Using the ergodic hypothesis and assuming a normalized molecule detection function ($V_{\text{eff}} = \int d^3 r \, \Omega(\mathbf{r}) = 1$), the concentration computes to $C = \langle F(t) \rangle / q$.

2.1.2 General Autocorrelation function for multiple species

Most experiments include particles with more than one dynamic property. Labeled particles may have different size or the temporal dynamics may include a triplet term. For n different species inside the detection volume, the autocorrelation function becomes:

$$G(\tau) = \frac{g(\tau)}{\langle F(t) \rangle^2} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij}(\tau)}{\langle F(t) \rangle^2} \tag{3}$$

$$g_{ij}(\tau) = q_i q_j \int \mathrm{d}^3 r \int \mathrm{d}^3 r' \, \Omega(\mathbf{r}) \Phi_{ij}(\mathbf{r}, \mathbf{r}', \tau) \Omega(\mathbf{r}') \tag{4}$$

$g(\tau)$ non normalized correlation function

$g_{ij}(\tau)$ non normalized cross correlation between two species i and j. For n species, $i, j \in [1, ..., n]$.

q_i molecular brightness of species i

Ω 3D molecule detection function

Φ_{ij} diffusion propagator computed from species i with species j. If species i and j are independently diffusing, then Φ_{ij} is zero. $C_{ij} \Phi_{ij}(\mathbf{r}, \mathbf{r}', \tau) = \langle \delta c_i(\mathbf{r}, 0) \delta c_j(\mathbf{r}', \tau) \rangle$

C_{ij} average concentration of objects following the dynamics of Φ_{ij}. If $i = j$, $C_{ii} = C_i$ is the concentration of the dye i.

If the propagators $\Phi_{ij}(x, y, z; x', y', z'; \tau)$ and the molecule detection function $\Omega(x, y, z)$ factorize into an axial (z) and a lateral (x, y) part, so will $g_{ij}(\tau)$:

$$g_{ij}(\tau) = q_i q_j \cdot g_{ij,z}(\tau) \cdot g_{ij,xy}(\tau) \tag{5}$$

Following the example with a freely diffusing species A and a laterally diffusing species B inside a membrane at $z = z_0$, it can be concluded:

$$\begin{aligned} g_{AA}(\tau) = &\quad q_A^2 \cdot g_{AA,z}(\tau) \cdot g_{AA,xy}(\tau) \\ g_{BB}(\tau) = &\quad q_B^2 \cdot g_{BB,z_0}(\tau) \cdot g_{BB,xy}(\tau) \\ g_{AB}(\tau) = g_{BA}(\tau) = &\quad q_A q_B \cdot g_{AB,z}(\tau) \cdot g_{AB,xy}(\tau) \\ g(\tau) = &\quad g_{AA}(\tau) + 2g_{AB}(\tau) + g_{BB}(\tau) \end{aligned}$$

To obtain the normalized autocorrelation function, the average $\langle F(t)\rangle$ has to be calculated:

$$\begin{aligned} F(t) = &\quad \sum_{i=1}^{n} F_i(t) \\ F_A(t) = &\quad q_A \int \mathrm{d}^3 r \, \Omega(\mathbf{r}) C_A(\mathbf{r}, t) \\ F_B(t) = &\quad q_B \int \mathrm{d}x \int \mathrm{d}y \, \Omega(x, y, z = z_0) C_B(x, y, t) \\ \langle F(t)\rangle = &\quad \langle F_A(t)\rangle + \langle F_B(t)\rangle \end{aligned}$$

It is noticeable, that C_B is a 2D concentration, whereas C_A is a 3D concentration. Since there is no correlation between the two freely diffusing species A and B, $g_{AB}(\tau)$ is zero. The normalized autocorrelation curve may now be calculated like this:

$$\begin{aligned} G(\tau) = &\quad \frac{g(\tau)}{\langle F(t)\rangle^2} \\ G(\tau) = &\quad \frac{g_{AA}(\tau) + g_{BB}(\tau)}{(\langle F_A(t)\rangle + \langle F_B(t)\rangle)^2} \end{aligned}$$

2.1.3 Cross-correlation

Cross-correlation is a generalization of autocorrelation. Cross-correlation functions are derived in the same manner as autocorrelation functions. Here, signals recorded in two detection channels are cross-correlated to obtain the correlation function.

$$G_{XY}(\tau) = \frac{\langle \delta F_X(t) \delta F_Y(t+\tau)\rangle}{\langle F_X(t)\rangle \langle F_Y(t)\rangle} \tag{6}$$

A cross-correlation analysis of two species labeled by two types of dyes observed in two corresponding detection channels can be used for binding assays. Only complexes giving simultaneous signal in both channels contribute to the cross-correlation amplitude. Thus a finite cross-correlation indicates co-diffusion.

2.1.4 Extension of the theory

By modifying the propagator Φ and the detection volume Ω, other effects, like triplet blinking or binding reactions can be quantified. In many cases, analytical solutions to the above integrals are not straightforward and approximations have to be made. For example, the Gaussian shaped detection profile in confocal FCS is already an approximation. However, deviations from the true results are considered to be small [16]. Section 3.3 introduces several model functions with various detection symmetries and particle dynamics.

2.2 Non-linear least-squares fit

PyCorrFit uses the non-linear least-squares fitting capabilities from `scipy.optimize`. This package utilizes the Levenberg–Marquardt algorithm to minimize the sum of the squares. More information on this topic can be obtained from the online documentation of `leastsq`[4]. 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:

$$d(G, H) = \sum_{i=1}^{n} [G(\tau_i) - H(\tau_i)]^2 \tag{7}$$

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

$$\chi^2 = \min_{\alpha_1,\dots,\alpha_k} \sum_{i=1}^{n} [G(\tau_i, \alpha_1, \dots, \alpha_k) - H(\tau_i)]^2 \tag{8}$$

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.

2.3 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:

$$\chi^2_{\text{weighted}} = \min_{\alpha_1,\dots,\alpha_k} \sum_{i=1}^{n} \frac{[G(\tau_i, \alpha_1, \dots, \alpha_k) - H(\tau_i)]^2}{\sigma_i^2} \tag{9}$$

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 3.1.

3 Working with PyCorrFit

3.1 Workflow

Figure 1 shows the main window of PyCorrFit. It contains a menu bar to access all tools, a notebook with tabs, each tab representing a single curve, and a page - the content of the currently selected tab.

3.1.1 A First Look

PyCorrFit is operated through the menu bar. The `File` menu lets the user load and save files as discussed in section 3.2.1. The `Tools` menu contains several tools that are described below. The `Current Page` menu allows the user to perform tasks on the current page. This includes loading experimental data files or saving the currently viewable correlation curve

[4] http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.leastsq.html#scipy.optimize.leastsq

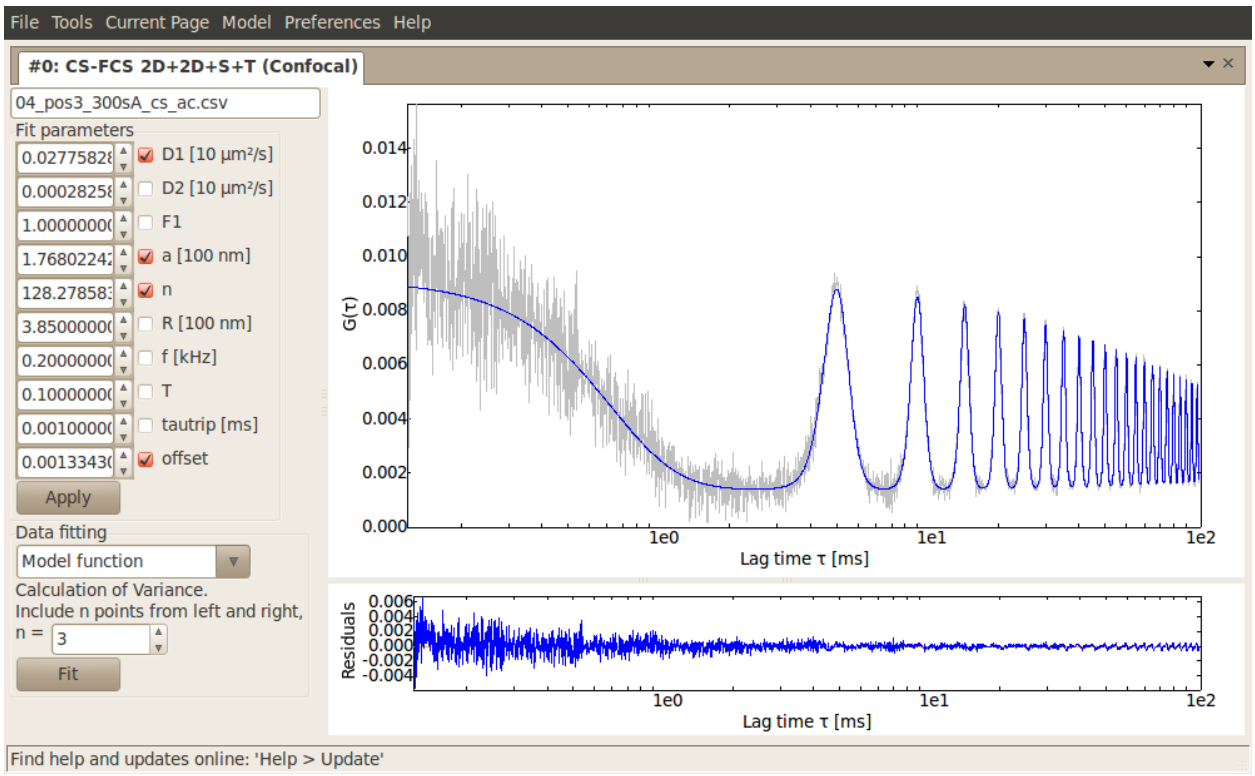

Figure 1, user interface of PyCorrFit: A circular scanning FCS (CS-FCS) curve of DiO on a supported lipid bilayer (glass substrate) is shown. The measurement yields a diffusion coefficient of 0.28 $\mu m^2 s^{-1}$ ($F1 = 1$, so only one component is fitted). Note that a 2D diffusion model is used and not a 3D model (as shown in figure 2).

as ~.csv file or as an image. The `Model` menu lets the user select a model to create a new page. The `Preferences` menu allows the user to use Latex for plotting (section 1.2.2) or tells the program to be more verbose (spline fits for calculation of weights will be displayed). The `Help` menu contains an update function that checks the current program version against the official release of PyCorrFit. The graphical user interface comes with a bunch of tools located in the `Tools` menu. The `Tools` menu will become active, once a page is added to the notebook. Each tool opens in a new window. Tools are updated or closed upon change of the current page.

3.1.2 Loading Experimental Data

Via `File - Load data files` experimental data files are opened and displayed on tab page. Within the page, the user may choose starting parameters for the fitting, select which parameters should be variable during fitting (checked means variable) and select a weighting method. The page displays the experimental and fitted curve, as well as the residuals in the lower plot.

3.1.3 Data Range Selection

This selects the lag times τ from the experimental curve that will be displayed. By default, all data from a file will be displayed.

3.1.4 Background Correction

Performs a background correction according to [12]. The result is a higher amplitude of the correlation fuction $G(0)$. Backgrounds may be manually set or imported from background measurements (see supported file formats).

$$G_{\text{corrected}}(\tau) = G_{\text{measured}}(\tau) \cdot \left[\frac{S}{S-B}\right]^2 \tag{10}$$

where S is the average signal of the correlation curve and B is the average background signal.

3.1.5 Performing a (weighted) Fit

Weights can be calculated for each data point using the standard deviation from neighboring points. Since FCS curves naturally have regions where the standard deviation of the experimental data is large (falling inflection points), the standard deviation has to be calculated from curves that mimic the experimental data. If weighted fitting is used, the weighted residuals are shown instead of the plain residuals (difference of experimental and theoretical curve). The following methods for weighting have been implemented:

Spline. A spline is fitted to the experimental curve. Weights are calculated from the deviation of the experimental data from the spline. If the `Verbose mode` is enabled in the `Preferences` menu, the spline fit will be displayed. The number of knots of the spline can be changed by adjusting the number in the dropdown menu. In case of multiple component data, a higher number of knots is required to obtain a fitting spline. For single component data, up to three knots are sufficient.

Model function. Instead of calculating the weights from a spline, calculate the weights from the previously fitted model function. This method of weighting is iterative - the fit should converge!

Average. If an average was created using the `Average data` tool, the standard deviation σ_i at all data points i, calculated from all curves can be used as weights for fitting.

3.1.6 Overlay Tool: Selecting valid Runs

Display all curves and easily remove outliers by deselecting them.

3.1.7 Batch Control: Fitting Several Curves at Once

Perform fitting operations on all pages. Parameters may be imported from other ~.fcsfit-session.zip-files. Application of parameters and fitting of curves is only executed for pages with the same model as the current page.

3.1.8 Global Fitting: When Curves Share Parameters

If multiple correlation curves share some of their parameters, a global fit can be applied. The `Overlay tool` may be used to select a subset of curves. Selected correlation curves are added to one array and a least-sqares fit is performed using the parameters the user chooses within the respective pages. For example, this is used in two-focus FCS, where two autocorrelation curves and two cross-correlation curves are obtained. These share the parameters particle number n and diffusion time τ_{diff}. A global fit can be applied such that n and τ_{diff} are identical for all curves.

3.1.9 Average Data: Obtaining an Average of all Runs

Averaging of curves within the session. The `Overlay tool` may be used to select a subset of curves. The tool asks if only pages with the same model as the current page should be used. Pages that do not have the same correlation type (autocorrelation or cross-correlation) as the current page are not used. The averaging does not work for data where the length of the correlation curves are not identical. Averaging in this case is not well-defined. For averaging, please make sure all curves in a series have the same measurement time. A new page will be created containing the resulting average. The resulting trace is constructed by appending the traces of the averaged curves. After averaging, the standard deviaton from all curves can be used as weights for fitting.

3.1.10 Slider Simulation: Which Parameter does What?

This tool is useful to visualize the impact of certain parameters on the shape of the correlation function or to set proper starting parameters for a fit.

3.1.11 Page Info: a Summary for each Curve

A most verbose information window that shows all there is to know about the current page. All information that is displayed here will also be saved when exporting a curve as a ~.csv file with the `Current Page` menu.

3.1.12 Trace View: Verification and Troubleshooting

If available, view the trace corresponding to the data of the current page. In case of cross-correlation, two traces will be displayed.

3.1.13 Statistics: Excel-ready Export of Results

Used for large data sets. Save selected parameters (of all pages with the same model as the current page) as a table of tab separated values into one single file.

3.2 Data file formats

3.2.1 Import: Support for common FCS file formats

PyCorrFit supports numerous data file formats. If a file format is not listed here, there are two options to make them work with PyCorrFit:

- Conversion to the native PyCorrFit ~.csv file format.
- Implementation of the file format within the `readfiles` module of PyCorrFit. `__init__.py` has to be edited and a script `read_FileFormat.py` has to be added.

Supported file formats:

~.asc files are created by e.g. the ALV-6000 hardware correlator.

~.csv files are the native PyCorrFit data files. See 3.2.3 for more information.

~.fcs files are created by Zeiss' ConfoCor3 software (e.g. AIM4.2 or ZEN2010b).

~.mat files are files generated by a Matlab program written by Jonas Ries. This was implemented for convenience.

~.sin files are created by the Flex correlators from correlator.com.

~.zip files may contain any of the files listed here.

3.2.2 Export and More: Interaction with the User

These are file formats that were designed for PyCorrFit.

3.2.3 .csv data files

The ~.csv file format contains experimental correlation curves and optionally the fitted curve, the residuals, as well as weighting factors that have been used for fitting. The data are stored as tab-separated values. PyCorrFit uses this format to export single correlation curves. The ~.csv files are directly readable by any editor on any platform, allow comments throughout the file and have a well-defined structure. There is a certain syntax that has to be followed for ~.csv files to be successfully imported by PyCorrFit:

- The **encoding** is preferably UTF-8. However, since no special characters are needed to save experimental data, any encoding might work. New line characters are `\r\n` (Windows).
- **Comments**: Lines starting with a hash (`#`) as well as empty lines or lines containing only white space characters are ignored. Exceptions are the keywords listed below.
- **Units**: A ~.csv file may contain one correlation curve, as well as up to two intensity traces. The time unit is seconds and the count rate (intensity) is measured in kHz. Correlation curves are assumed to be calculated from fluctuations around zero, thus the correlation curve has to decrease to zero at large lag times.
- The first table inside a ~.csv file contains the **correlation curve**. The table consists of either two or four columns of tab-separated values. The first column contains the lag times in seconds and the second column contains the correlation curve. PyCorrFit also saves the fitted curve and the residuals in the third and fourth column, as well as optional weighting factors that were used for fitting. This additional information is ignored during file import.
- **First trace**: The table containing the correlation curve is considered to have ended as soon as the keyword `# BEGIN TRACE` appears at the beginning of a line. Then, the trace will be read, again as tab-separated values: The first column contains the time in seconds and second column contains the corresponding signal in KHz.
- The optional **second trace** is initiated with the keyword `# BEGIN SECOND TRACE`.
- **AC/CC**: To distinguish the types of imported experimental curves (autocorrelation or cross-correlation), the keyword `Data type` can be used. If a line starts with either of the following:

```
# Data type Cross-correlation
# Data type Autocorrelation
```

then the corresponding string `AC` or `CC` will be used to identify the type of the curve in the user interface. If no data type is specified, an autocorrelation curve is assumed. Curves may be personalized by using:

```
# Data type Autocorrelation _A1
```

This will result in the identifier `AC_A1` during import in the user interface. This feature is useful to e.g. distinguish many ~.csv files of different type inside a ~.zip file.

3.2.4 .zip data files

~.zip files are either used as session files or as containers for bundled correlation curves. **~.fcsfit-session.zip files** are used to store entire PyCorrFit sessions including all parameters and modifications. They can be imported as sessions or as experimental curves. **~.zip files** are containers for files containing correlation curves.

3.2.5 .txt model function files

In addition to the built-in model functions, PyCorrFit supports the import of user defined model functions for fitting. To get started quickly, it is helpful to take a look at some examples online[5]. File syntax:

- **Parameters**: PyCorrFit works with the following dimensional representation. Any other unit has to be derived from these:
  - unit of time : 1 ms
  - unit of inverse time: $1000\,\mathrm{s}^{-1}$
  - unit of distance : 100 nm
  - unit of diffusion coefficient : $10\,\mu\mathrm{m}^2\mathrm{s}^{-1}$
  - unit of inverse area: $100\,\mu\mathrm{m}^{-2}$
  - unit of inverse volume : $1000\,\mu\mathrm{m}^{-3}$
- The **encoding** is UTF-8. On Windows, `Notepad++` can be used. Under Linux, editors like `gedit` or `kate` can be used.
- **Comments**: Lines starting with a hash (`#`) (except the first line) as well as empty lines or lines containing only white space characters are ignored.
- **Name**: The first line contians the name of the correlation function, preceded by a hash, e.g.
  `# AC Gauss test`.
- **Definition of parameters** To define parameters used by the model function, they have to be named and given a starting value. Optionally a dimension of the parameter may be added before the equal sign and separated from the parameter name by a white space. The dimension is only displayed by PyCorrFit and not processed, e.g.
  `D [10 μm²/s] = 200.0`.
  This is done for all parameters of the model function. Parameters may not start with a `G` or `g`. Also, some variables like `e` or `pi` are already mapped and must not be used. The parameter `tau` is reserved for the corresponding lag times (ms) and may be used freely.

[5] "PyCorrFit external model functions" in the download section at http://pycorrfit.craban.de

- **Placeholders:** Placeholders must start with a lower `g`, for example `gTwoD = 1./(1.+D*tau/a**2)`.
- **Correlation function**: The correlation function is identified by the capital letter `G`: `G = 1./n * gThrD * gScan * gTrip`

Figure 2 shows a working example. External models will be imported with internal model function IDs starting at 7000. Model functions are checked upon import by PyCorrFit. If the model function does not work, it might be a syntax error, or just an error of `sympy`. Sympy is used to check the model and is still under development. Model functions will be saved upon session saving.

3.3 Implemented model functions

This is an overview of all the model functions that are currently[6] implemented in PyCorrFit. To each model a unique model ID is assigned by PyCorrFit. Most of the following information is also accessible from within PyCorrFit using the **Page info** tool.

3.3.1 Confocal FCS

The confocal detection volume with the structural parameter

$$SP = \frac{z_0}{r_0} \tag{11}$$

has an effective size of

$$V = \pi^{3/2} r_0^2 z_0 \tag{12}$$

where r_0 is its lateral and z_0 its axial (in case of 3D diffusion) extension. Thus, the effective number of particles is defined as

$$N = CV \tag{13}$$

with the concentration C given implicitly in the model functions. The diffusion coefficient is calculated from the diffusion time τ_{diff} using

$$D = \frac{1}{4\tau_{\mathrm{diff}}} \left(\frac{z_0}{SP}\right)^2 = \frac{r_0^2}{4\tau_{\mathrm{diff}}}. \tag{14}$$

The parameters in the equation above need to be calibrated to obtain the diffusion coefficient. Usually a reference dye with a known diffusion coefficient is used to determine the lateral extension of the detection volume r_0 with a fixed structural parameter of e.g. $SP = 4$.

Name	**2D (Gauß)**
ID	**6001**
Descr.	Two-dimensional diffusion with a Gaussian laser profile [1,8,10].

$$G(\tau) = A_0 + \frac{1}{N} \frac{1}{(1 + \tau/\tau_{\mathrm{diff}})} \tag{15}$$

[6] October 28, 2013

```
# CS-FCS 3D+S+T (Confocal)

# Circular Scanning FCS model function. 3D diffusion + Triplet.

## 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
D [10 μm^2/s] = 200.0
# Structural parameter
w = 5.0
# Waist of the lateral detection area
a [100 nm] = 1.0
# Particle number
n = 5.0
# Scan radius
R [100 nm] = 5.0
# Frequency
f [kHz] = 20.0
# Triplet fraction
T = 0.1
# Triplet time
tautrip [ms] = 0.001

# 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.

gTrip = 1. + T/(1-T)*exp(-tau/tautrip)
gScan = exp(-(R*sin(pi*f*tau))**2/(a**2+D*tau))
gTwoD = 1./(1.+D*tau/a**2)
gOneD = 1./sqrt(1.+D*tau/(w*a)**2)
gThrD = gTwoD * gOneD

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

G = 1./n * gThrD * gScan * gTrip
```

Figure 2, user defined model function for PyCorrFit: The working example shows a model function for circular scanning FCS.

A_0	Offset
N	Effective number of particles in confocal area
τ_{diff}	Characteristic residence time in confocal area

Name **2D+T (Gauß)**
ID **6002**
Descr. Two-dimensional diffusion with a Gaussian laser profile, including a triplet component [1, 5, 8, 10, 13, 14].

$$G(\tau) = A_0 + \frac{1}{N}\frac{1}{(1+\tau/\tau_{\text{diff}})}\left(1 + \frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{16}$$

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 **3D (Gauß)**
ID **6012**
Descr. Three-dimensional free diffusion with a Gaussian laser profile (eliptical) [1, 8, 10].

$$G(\tau) = A_0 + \frac{1}{N}\frac{1}{(1+\tau/\tau_{\text{diff}})}\frac{1}{\sqrt{1+\tau/(SP^2\tau_{\text{diff}})}} \tag{17}$$

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

Name **3D+T (Gauß)**
ID **6011**
Descr. Three-dimensional free diffusion with a Gaussian laser profile (eliptical), including a triplet component [5, 13, 14].

$$G(\tau) = A_0 + \frac{1}{N}\frac{1}{(1+\tau/\tau_{\text{diff}})}\frac{1}{\sqrt{1+\tau/(SP^2\tau_{\text{diff}})}}\left(1 + \frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{18}$$

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 **2D+2D+T (Gauß)**
ID **6031**
Descr. Two-component, two-dimensional diffusion with a Gaussian laser profile, including a triplet component [1,3,7,12].

$$G(\tau) = A_0 + \frac{1}{N(F+\alpha(1-F))^2}\left[\frac{F}{1+\tau/\tau_1} + \alpha^2\frac{1-F}{1+\tau/\tau_2}\right]\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{19}$$

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 **3D+2D+T (Gauß)**
ID **6032**
Descr. Two-component, two- and three-dimensional diffusion with a Gaussian laser profile, including a triplet component [1,3,7,12].

$$G(\tau) = A_0 + \frac{1}{N(1-F+\alpha F)^2}\left[\frac{1-F}{1+\tau/\tau_{2D}} + \frac{\alpha^2 F}{(1+\tau/\tau_{3D})}\frac{1}{\sqrt{1+\tau/(SP^2\tau_{3D})}}\right]\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right) \tag{20}$$

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

Name **3D+3D+T (Gauß)**
ID **6030**
Descr. Two-component three-dimensional free diffusion with a Gaussian laser profile, including a triplet component [1,3,7,12].

$$\begin{aligned} G(\tau) = A_0 + \frac{1}{N(F+\alpha(1-F))^2}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\times \\ \times\left[\frac{F}{(1+\tau/\tau_1)}\frac{1}{\sqrt{1+\tau/(SP^2\tau_1)}} + \alpha^2\frac{1-F}{(1+\tau/\tau_2)}\frac{1}{\sqrt{1+\tau/(SP^2\tau_2)}}\right] \end{aligned} \tag{21}$$

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

3.3.2 Confocal TIR-FCS

The detection volume is axially confined by an evanescent field and has an effective size of

$$V = \pi R_0^2 d_{\text{eva}} \tag{22}$$

where R_0 is the lateral extent of the detection volume and d_{eva} is the evanescent field depth[7]. From the concentration C, the effective number of particles is $N = CV$. The decay constant κ is the inverse of the depth d_{eva} :

$$d_{\text{eva}} = \frac{1}{\kappa} \tag{23}$$

The model functions make use of the Faddeeva function (complex error function)[8]:

$$\begin{aligned} w(i\xi) &= e^{\xi^2}\operatorname{erfc}(\xi) \\ &= e^{\xi^2} \cdot \frac{2}{\sqrt{\pi}} \int_\xi^\infty \mathrm{e}^{-\alpha^2}\mathrm{d}\alpha \end{aligned} \tag{24}$$

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.

Name **3D (Gauß/exp)**
ID **6013**
Descr. Three-dimensional free diffusion with a Gaussian lateral detection profile and an exponentially decaying profile in axial direction [4,6,11].

$$G(\tau) = \frac{1}{C}\frac{\kappa^2}{\pi(R_0^2 + 4D\tau)}\left(\sqrt{\frac{D\tau}{\pi}} + \frac{1 - 2D\tau\kappa^2}{2\kappa} w\left(i\sqrt{D\tau}\kappa\right)\right) \tag{25}$$

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

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

[8] In user-defined model functions, 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.

Name **3D+2D+T (Gauß/exp)**
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 [1,3,4,6,7,11,12].

$$G(\tau) = A_0 + \frac{1}{N(1-F+\alpha F)^2}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\times \tag{26}$$
$$\times\left[\frac{1-F}{1+4D_{2\text{D}}\tau/R_0^2}+\frac{\alpha^2 F\kappa}{1+4D_{3\text{D}}\tau/R_0^2}\left(\sqrt{\frac{D_{3\text{D}}\tau}{\pi}}+\frac{1-2D_{3\text{D}}\tau\kappa^2}{2\kappa}w\left(i\sqrt{D_{3\text{D}}\tau}\kappa\right)\right)\right]$$

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

Name **3D+3D+T (Gauß/exp)**
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 [1,3,4,6,7,11,12].

$$G(\tau) = A_0 + \frac{1}{N(1-F+\alpha F)^2}\left(1+\frac{Te^{-\tau/\tau_{\text{trip}}}}{1-T}\right)\times \tag{27}$$
$$\times\left[\frac{F\kappa}{1+4D_1\tau/R_0^2}\left(\sqrt{\frac{D_1\tau}{\pi}}+\frac{1-2D_1\tau\kappa^2}{2\kappa}w\left(i\sqrt{D_1\tau}\kappa\right)\right)+\right.$$
$$\left.+\frac{(1-F)\alpha^2\kappa}{1+4D_2\tau/R_0^2}\left(\sqrt{\frac{D_2\tau}{\pi}}+\frac{1-2D_2\tau\kappa^2}{2\kappa}w\left(i\sqrt{D_2\tau}\kappa\right)\right)\right]$$

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

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

The detection volume is axially confined by an evanescent field of depth[9] $d_{\mathrm{eva}} = 1/\kappa$. 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. The model functions make use of the Faddeeva function (complex error function)[10]:

$$\begin{aligned} w(i\xi) &= e^{\xi^2}\operatorname{erfc}(\xi) \\ &= e^{\xi^2} \cdot \frac{2}{\sqrt{\pi}} \int_{\xi}^{\infty} \mathrm{e}^{-\alpha^2} \mathrm{d}\alpha \end{aligned} \tag{29}$$

Name **2D (□xσ)**
ID **6000**
Descr. Two-dimensional diffusion with a square-shaped lateral detection area taking into account the size of the point spread function [9, 15][11].

$$G(\tau) = \frac{1}{C}\left[\frac{2\sqrt{\sigma^2 + D\tau}}{\sqrt{\pi}a^2}\left(\exp\left(-\frac{a^2}{4(\sigma^2 + D\tau)}\right) - 1\right) + \frac{1}{a}\operatorname{erf}\left(\frac{a}{2\sqrt{\sigma^2 + D\tau}}\right)\right]^2 \tag{30}$$

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 **3D (□xσ/exp)**
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 [9, 15].

$$\begin{aligned} G(\tau) = \frac{\kappa^2}{C}&\left(\sqrt{\frac{D\tau}{\pi}} + \frac{1 - 2D\tau\kappa^2)}{2\kappa} w\left(i\sqrt{D\tau}\kappa\right)\right) \times \\ &\times\left[\frac{2\sqrt{\sigma^2 + D\tau}}{\sqrt{\pi}a^2}\left(\exp\left(-\frac{a^2}{4(\sigma^2 + D\tau)}\right) - 1\right) + \frac{1}{a}\operatorname{erf}\left(\frac{a}{2\sqrt{\sigma^2 + D\tau}}\right)\right]^2 \end{aligned} \tag{31}$$

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 ($\kappa = 1/d_{\mathrm{eva}}$)
D	Diffusion coefficient

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

[10] In user-defined model functions, 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.

Name **2D+2D (□xσ/exp)**
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) [9, 15].

Name **3D+2D (□xσ/exp)**
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) [9, 15].

Name **3D+3D (□xσ/exp)**
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) [9, 15].

Name **3D+2D+kin (□xσ/exp)**
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 [9]. Because approximations are made in the derivation, please verify if this model is applicable to your problem before using it.

Acknowledgements

I thank André Scholich (TU Dresden, Germany) for initial proof reading of the manuscript and Grzegorz Chwastek, Franziska Thomas, and Thomas Weidemann (Biotec, TU Dresden, Germany) for critical feedback on PyCorrFit.

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. Available from: http://link.aip.org/link/?JCP/64/1791/1, doi:10.1063/1.432357.

[2] Markus Burkhardt. *Electron multiplying CCD – based detection in Fluorescence Correlation Spectroscopy and measurements in living zebrafish embryos*. PhD thesis, Biophysics, BIOTEC, Technische Universität Dresden, Tatzberg 47–51, 01307 Dresden, Germany, 2010. http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-61021.

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

[4] 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. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0006349505730794, doi:10.1529/biophysj.104.053884.

[5] 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. Available from: http://www.pnas.org/content/95/23/13573.abstract, arXiv:http://www.pnas.org/content/95/23/13573.full.pdf+html, doi:10.1073/pnas.95.23.13573.

[6] 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. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1614500&tool=pmcentrez&rendertype=abstract, doi:10.1529/biophysj.105.074625.

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

[8] 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. Available from: http://ao.osa.org/abstract.cfm?URI=ao-30-10-1185, doi:10.1364/AO.30.001185.

[9] Jonas Ries, Eugene P. Petrov, and Petra Schwille. Total internal reflection fluorescence correlation spectroscopy: Effects of lateral diffusion and surface-generated fluorescence. *Biophysical Journal*, 95(1):390 – 399, 2008. Available from: http://www.sciencedirect.com/science/article/pii/S0006349508703126, doi:10.1529/biophysj.107.126193.

[10] 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. Available from: http://dx.doi.org/10.1007/BF00185777, doi:10.1007/BF00185777.

[11] 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. Available from: http://www.sciencedirect.com/science/article/pii/S0006349501761309, doi:10.1016/S0006-3495(01)76130-9.

[12] Nancy Thompson. Fluorescence correlation spectroscopy. In Joseph Lakowicz, Chris D. Geddes, and Joseph R. Lakowicz, editors, *Topics in Fluorescence Spectroscopy*, volume 1 of *Topics in Fluorescence Spectroscopy*, pages 337–378. Springer US, 2002. Available from: http://dx.doi.org/10.1007/0-306-47057-8_6, doi:10.1007/0-306-47057-8_6.

[13] 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. Available from: http://pubs.acs.org/doi/abs/10.1021/j100036a009, arXiv:http://pubs.acs.org/doi/pdf/10.1021/j100036a009, doi:10.1021/j100036a009.

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

[15] 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. Available from: http://link.aip.org/link/?RSI/82/036105/1, doi:10.1063/1.3557412.

[16] Bo Zhang, Josiane Zerubia, and Jean-Christophe Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics*, 46(10):1819–1829, Apr 2007. Available from: http://ao.osa.org/abstract.cfm?URI=ao-46-10-1819, doi:10.1364/AO.46.001819.