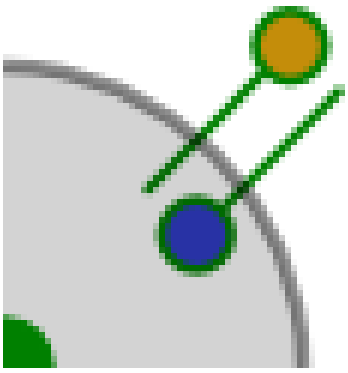
PyScanFCS Documentation

Release 0.3.0

Paul Müller

May 29, 2018

Contents:



PyScanFCS

Welcome to the documentation of PyScanFCS (version 0.3.0), a data analysis software for perpendicular line scanning fluorescence correlation spectroscopy.

CHAPTER 1

About PyScanFCS

When a membrane is scanned perpendicularly to its surface, the fluorescence signal originating from the membrane itself must be separated from the signal of the surrounding medium for an FCS analysis. PyScanFCS interactively extracts the fluctuating fluorescence signal from such measurements and applies a multiple-tau algorithm. The obtained correlation curves can be evaluated using PyCorrFit.

1.1 Supported operating systems

- Windows 7 and higher
- Linux (Debian-based)
- Any other operating system with a Python 3.6 installation (via pip)

1.2 Supported filetypes

- Correlator.com Flex correlator (~.dat - photon stream data)

1.3 How to cite

Müller, P., Schwille, P., and Weidemann, T. Fluorescence Spectroscopy and Microscopy, chapter: *Scanning Fluorescence Correlation Spectroscopy (SFCS) with a Scan Path Perpendicular to the Membrane Plane*. Methods in Molecular Biology, Humana Press, Springer, New York. 2014, 635-651. doi:10.1007/978-1-62703-649-8_29 - download author's manuscript.

CHAPTER 2

Gallery

TODO:

- Automatically insert all images from the gallery folder.
- Add new screenshots.

CHAPTER 3

Getting started

3.1 Installation

- Windows installers for PyScanFCS are available at the release page. Note: The installer is currently broken because astropy cannot be frozen, see https://github.com/astropy/astropy/issues/7052.
- On Debian-based systems, install via `apt-get install pyscanfcs`.
- If you have Python 3.6 installed, you may install PyScanFCS via `pip install pyscanfcs[GUI]`. After the installation, type `pyscanfcs` in a command shell to start PyScanFCS.

3.2 Documentation

The documentation is in the process of being transferred entirely to readthedocs.org. Currently, the it is scattered across several places and it is most-likely outdated:

- Original LaTeX-based PDF file (outdated): https://github.com/FCS-analysis/PyScanFCS/wiki/PyScanFCS_doc.pdf
- Wiki pages: https://github.com/FCS-analysis/PyScanFCS/wiki

CHAPTER 4

Contribute

4.1 General remarks

PyScanFCS has no funding and no developer community. My personal objective is to keep PyScanFCS operational on Linux and Windows which is currently limited by the free time I have available.

An active community is very important for an open source project such as PyScanFCS. You can help this community grow (and thus help improve PyScanFCS) in numerous ways:

1. Tell your colleagues and peers about PyScanFCS. One of them might be able to contribute to the project.
2. If you need a new feature in PyScanFCS, publicly announce a bounty for its implementation.
3. If your research heavily relies on FCS, please consider diverting some of your resources to the development of PyScanFCS.
4. You don't have to be a Python programmer to contribute. If you are familiar with reStrucuredText or LaTeX, you might be able to help out with the online documentation.
5. Please cite: Müller et al. Methods in Molecular Biology, Humana Press, Springer, New York. 2014, 635-651. doi:10.1007/978-1-62703-649-8_29

If you are planning to contribute to PyScanFCS, please contact me via the PyScanFCS issue page on GitHub such that we may coordinate a pull request.

4.2 For documentation writers

To build this documentation, fork PyScanFCS, navigate to the *docs* (not *doc*) directory and run.

- `pip install -r requirements.txt` followed by
- `sphinx-build . _build`.

This will create the html documentation on your computer. Syntax warnings and errors will be displayed during the build (there should be none). After making your changes to your forked branch, create a pull request on GitHub.

If you only found a typo or wish to make text-only changes, you can also use the GitHub interface to edit the files (without testing the build step on your computer).

4.3 For developers

4.3.1 Running from source

It is recommended to work with virtual environments.

Windows

The easiest way to run PyScanFCS from source on Windows is Anaconda.

- `conda install matplotlib numpy pip scipy wxpython`
- `pip install cython wheel simplejson sympy lmfit`
- `pip install -e . # in the root directory of the repository`

Ubuntu 17.10

PyScanFCS requires wxPython >= 4.0.1 which is not available as a binary wheel on PyPI and thus must be built from the .tar.gz. Install all dependencies (https://github.com/wxWidgets/Phoenix/blob/master/README.rst):

- `pip install cython matplotlib lmfit numpy scipy sympy`
- `sudo apt-get install -qq libgtk2.0 libgtk2.0-dev libwebkitgtk-dev dpkg-dev build-essential python3.6-dev libjpeg-dev libtiff-dev libsdl1.2-dev libnotify-dev freeglut3 freeglut3-dev libsm-dev libgtk-3-dev libwebkit2gtk-4.0-dev libxtst-dev libgstreamer-plugins-base1.0-dev`
- `pip install wxPython # this will take some time`
- `pip install -e . # in the root directory of the repository`

4.3.2 Testing

PyScanFCS is tested using pytest. If you have the time, please write test methods for your code and put them in the `tests` directory. You may run all tests by issuing:

```
python setup.py test
```

4.3.3 Pull request guidelines

Please fork PyScanFCS and create a pull request (PR) introducing your changes.

- A new PR should always be made into the *master* branch.
- If a PR introduces a new functionality or fixes a bug, it should provide a test case, i.e. a new file or function in the *tests* directory (see here for examples). Note that currently there is no recipe for testing the graphical user interface code.
- New code should follow the style guide for Python. Please use flake8 to check the files you changed or created.

- New code should be documented well.
- Make sure to update the changelog.

4.3.4 Windows test binaries

After each commit to the PyScanFCS repository, a binary installer is created by Appveyor. Click on a build and navigate to `ARTIFACTS` (upper right corner right under the running time of the build). From there you can download the Windows installer for the commit.

CHAPTER 5

Changelog

List of changes in-between PyScanFCS releases.

5.1 version 0.3.0

- Move from Python 2 to Python 3 (#12)
- Move to wxPython 4.0.1 (#12)
- Move from deprecated pyfits to astropy (#10)
- Support numpy 1.13
- Fix two spelling mistakes (#11)
- Cython code cleanup using flake8
- Bugfix: trailing photon events were not used (<1% of input data)
- Remove manual update checker

5.2 version 0.2.3

- New tool to create artificial .dat files for testing
- A couple of minor bug fixes
- Exceptions are now handled by wxPython
- Under the hood:
  - Updated repository structure
  - Relative imports
  - Build machine is now Windows 7

5.3 version 0.2.2

- Added button 'full time interval'
- This is the first version that uses the external module *multipletau*
- This is the first version that is available at PyPI
- Added setup files with Cython support
- MakeTestDat_FCS.py can be used to create test data
- Tested with Mac OSx

5.4 version 0.2.1

- Updated help menu (documentation, wiki, update)
- Performed some artwork around PyScanFCS
- Output of bleaching correction data (#1)
- Code cleanup

5.5 version 0.2.0

- Code cleanup
- Considerable speed-up in binning of .dat files
- Secured compatibility with PyCorrFit (.csv files)
- Fixed small file-naming bug

5.6 version 0.1.9

- Code cleanup
- Moved to Python 2.7

5.7 version 0.1.8

- Initital GitHub commit

CHAPTER 6

Indices and tables

- genindex
- modindex
- search