

Systems biology

PyFDAP: automated analysis of fluorescence decay after photoconversion (FDAP) experiments

Alexander Bläßle and Patrick Müller*

Systems Biology of Development Group, Friedrich Miescher Laboratory of the Max Planck Society, 72076 Tübingen, Germany

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on August 18, 2014; revised on October 16, 2014; accepted on November 2, 2014

Abstract

Summary: We developed the graphical user interface PyFDAP for the fitting of linear and non-linear decay functions to data from fluorescence decay after photoconversion (FDAP) experiments. PyFDAP structures and analyses large FDAP datasets and features multiple fitting and plotting options.

Availability and implementation: PyFDAP was written in Python and runs on Ubuntu Linux, Mac OS X and Microsoft Windows operating systems. The software, a user guide and a test FDAP dataset are freely available for download from <http://people.tuebingen.mpg.de/mueller-lab>.

Contact: pmueller@tue.mpg.de

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Fluorescence microscopy techniques have been widely used to measure the properties of fluorescently labeled proteins in embryos, tissues and cells (Pantazis and Supatto, 2014). Over the last decade, the repertoire of fluorescent proteins has been extended with the discovery and engineering of photoconvertible proteins that change their emission properties upon light exposure and that can be used to track sub-populations of pulse-labeled proteins (Adam *et al.*, 2014).

Photoconvertible proteins have recently been adapted to determine protein stability via microscopy-based fluorescence decay after photoconversion (FDAP) assays [reviewed in Rogers *et al.* (2015)]. In FDAP assays, a protein of interest is tagged with a photoconvertible protein and expressed *in vivo*. After photoconversion, the decay in fluorescence intensity of the protein is monitored. The data is then fitted with a decay function that models protein clearance to calculate protein half-lives.

Protein stability is context-dependent. For example, protein stability might be different inside cells and in the extracellular space (Müller *et al.*, 2013; Rogers *et al.*, 2015; Zhou *et al.*, 2012). Both intra- and extracellular protein half-lives can be determined by FDAP using a static intracellular signal to create masks for intra- and extracellular intensities (Rogers *et al.*, 2015).

We have previously described how to execute FDAP experiments in living zebrafish embryos (Müller *et al.*, 2012; Rogers *et al.*, 2015). Here, we provide a standardized computational framework to analyse the resulting datasets. Our software PyFDAP features (i) a comprehensive data format for handling, sorting and annotating large FDAP datasets, (ii) the ability to separate FDAP datasets into their intra- and extracellular components based on counter-labeling, (iii) established fitting algorithms and (iv) a user-friendly environment that allows researchers from a non-computational background to easily evaluate FDAP datasets.

2 Implementation

PyFDAP was developed as an open-source graphical user interface (GUI) in Python with PyQT and SciPy to make it accessible across the most frequently used operating systems: Ubuntu Linux, Mac OS X and Microsoft Windows. Python is a widely used scientific programming language and provides PyFDAP users with enormous resources and easily incorporated software packages (Millman and Aivazis, 2011). The PyFDAP GUI was designed to enable the user to simultaneously view the current project tree, project properties and various tabbed plots.

Users import FDAP datasets as '.tif' files. As intra- and extracellular protein stabilities can be different, PyFDAP offers an option to import a second dataset that counter-labels intra- or extracellular space. The separation of fluorescence intensities into intra- or extracellular masks is performed using the Otsu binarization algorithm (Otsu, 1979). The masks and corresponding datasets can then be investigated inside the PyFDAP GUI. The masks are applied to the images of the photoconverted signal, and the average intensities in the intra- and extracellular domains and in the entire image are calculated. PyFDAP organizes all data into a logical hierarchical object structure to facilitate data navigation (Fig. 1).

PyFDAP offers multiple fitting options to fit an exponential decay function given by

$$c(t) = c_0 e^{-kt} + y_0$$

or an inverse power function given by

$$c(t) = (c_0^{1-n} - kt(1-n))^{\frac{1}{1-n}} + y_0$$

to the dataset to model linear (Müller *et al.*, 2012) or non-linear decay (Eldar *et al.*, 2003), respectively. Here, $c(t)$ is the intensity of the decaying fluorescent signal, y_0 the baseline of the decay, $c_0 + y_0$ the fluorescent intensity at $t=0$ and k the decay rate constant of the fluorescent signal. In the case of non-linear decay, n controls the degree of non-linearity. PyFDAP calculates the protein half-lives τ from k as $\tau = \ln(2)/k$ or $\tau = (2^{n-1} - 1)c_0^{1-n}/(k(n-1))$ for linear and non-linear decay, respectively.

PyFDAP offers several established bounded and unbounded optimization algorithms to fit the decay models to the data. The lower bound of fluorescence measurements is limited by the intrinsic background signal of the microscope and the autofluorescence of the imaged sample, and it is therefore crucial to define biologically reasonable bounds for the fit. PyFDAP offers an option to import background and noise measurements and allows the user to choose a lower bound of the baseline y_0 from four presets: (i) the level of noise N estimated from the noise dataset, (ii) the mean level of the pre-conversion background datasets \bar{B}_{pre} , (iii) the mean background

level over all background datasets \bar{B} or (iv) a special weighting function F (Müller *et al.*, 2012) given by

$$F_{i,r} = \frac{1}{b} \sum_{j=1}^b \min_t \left(\frac{B_{j,r}(t) - N_i}{B_{\text{pre},r} - N_i} \right),$$

where i is the current FDAP measurement, $r \in \{\text{intracellular, extracellular, entire domain}\}$ is the investigated region and $j \in \{1, \dots, b\}$ represents the indices of background datasets with intensities $B(t)$ at time t . Using the function F , users can compute the lower bound of the baseline for measurement i and region r , $y_{0,i,r}$, by

$$y_{0,i,r} \geq F_{i,r} \cdot (I_{\text{pre},r} - N_i) + N_i,$$

where $I_{\text{pre},r}$ denotes the pre-conversion intensity of the FDAP measurement i in region r . Together, these four presets offer a wide range of possibilities for the estimation of the baseline y_0 .

Multiple plotting options help the user to remove or improve corrupt datasets or fits. The user can plot the optimized parameters to find outliers and re-run the fits with different options. Moreover, PyFDAP saves the fitting progress and lets the user 'slide' through it to identify points at which optimization algorithms got arrested in a local minimum and where parameters need to be changed.

PyFDAP allows data and results to be saved in various image and video formats. Individual decay curves and fits can be exported as '.csv' files. Similarly, complete PyFDAP project trees (Fig. 1) including all settings and properties can be exported as '.csv' or JavaScript Object Notation (JSON) object files. Object files can easily be reloaded into PyFDAP, which ensures full reproducibility of data analysis and facilitates collaboration among researchers.

To offer customizability and debugging options, PyFDAP also comes with a full Python terminal that allows users to modify FDAP datasets or to improve FDAP data analysis without the need to edit the main PyFDAP code.

A detailed user guide (Supplementary Material) describes the installation and usage of the PyFDAP software as well as the mathematical background for data analysis.

3 Conclusion

PyFDAP is an open-source GUI for the analysis and handling of FDAP data. Usage of PyFDAP does not require any programming knowledge. The straightforward GUI provides a quick workflow, and the integrated terminal and open-source nature of PyFDAP enable researchers to extend the software to suit their needs.

Acknowledgements

We thank Katherine Rogers, Luciano Marcon, David Mörsdorf and Gary Soh for useful suggestions and discussions. This work was supported by the Emmy Noether Program of the Deutsche Forschungsgemeinschaft, the Max Planck Society and a Career Development Award from the Human Frontier Science Program.

References

- Adam, V. *et al.* (2014) Phototransformable fluorescent proteins: future challenges. *Curr. Opin. Chem. Biol.*, 20C, 92–102.
- Eldar, A. *et al.* (2003) Self-enhanced ligand degradation underlies robustness of morphogen gradients. *Dev. Cell*, 5, 635–646.
- Millman, K.J. and Aivazis, M. (2011) Python for scientists and engineers. *Comput. Sci. Eng.*, 13, 9–12.
- Müller, P. *et al.* (2012) Differential diffusivity of Nodal and Lefty underlies a reaction-diffusion patterning system. *Science*, 336, 721–724.

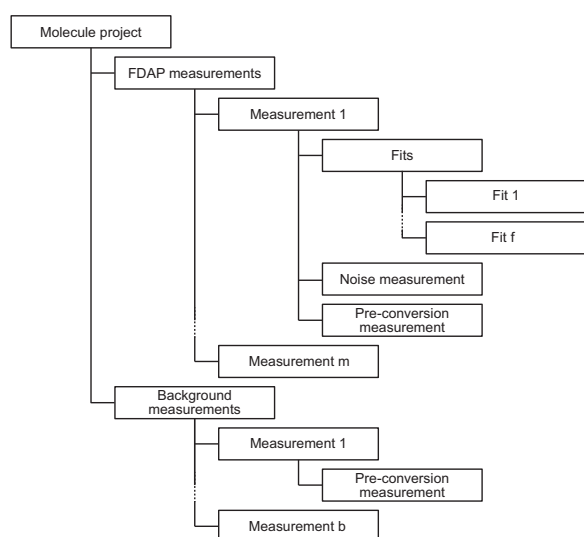


Fig. 1. Hierarchical PyFDAP data structure for FDAP experiments. Experiments are grouped into a main molecule project and divided into FDAP (1 to m) and background (1 to b) pre- and post-conversion measurements. Each FDAP measurement can have multiple fits (1 to f) with different fitting options

- Müller,P. *et al.* (2013) Morphogen transport. *Development*, **140**, 1621–1638.
- Otsu,N. (1979) A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.*, **9**, 62–66.
- Pantazis,P. and Supatto,W. (2014) Advances in whole-embryo imaging: a quantitative transition is underway. *Nat. Rev. Mol. Cell Biol.*, **15**, 327–339.
- Rogers,K.W. *et al.* (2015) Measuring protein stability in living zebrafish embryos using Fluorescence Decay After Photoconversion (FDAP). *J. Vis. Exp.*, [Epub ahead of print, doi: 10.3791/52266, January 28, 2015].
- Zhou,S. *et al.* (2012) Free extracellular diffusion creates the Dpp morphogen gradient of the *Drosophila* wing disc. *Curr. Biol.*, **22**, 668–675.