

Going beyond simplified models: Improving FRAP analysis using PyFRAP

Alexander Bläßle, Gary Soh, Patrick Müller

June 29, 2016

Abstract

The transport of macromolecules is crucial for development and homeostasis and must be tightly regulated to prevent developmental defects and disease states. Macromolecules such as proteins must often move through complex environments from their site of production to their targets. What are the biophysical properties of such molecules, and how is their transport affected by the properties of the environment through which they move? One technique to address these questions is FRAP (Fluorescence Recovery After Photobleaching), which measures the mobility-driven recovery of fluorescent molecules in a photobleached area. Although FRAP assays have been used for 40 years, there are several shortcomings of current data analysis methods that preclude the retrieval of quantitative information from FRAP experiments. Most analysis algorithms make simplified assumptions about the sample geometry, bleaching conditions, and the underlying kinetics of the system that do not fully reflect the complexities of the experimental sample. Furthermore, the algorithms used to analyze FRAP data cannot always be easily implemented in laboratories without the appropriate computing background. To address these shortcomings, we developed the Python software package “PyFRAP”. PyFRAP numerically solves the equations that govern macromolecule transport on an exact three-dimensional model of the sample using finite element simulations. To account for bleaching inhomogeneities, PyFRAP matches the FRAP image data with the initial condition on the simulation. Simulated solutions are then fitted to the experimental data using a variety of optimization algorithms. PyFRAP comes with an intuitive graphical user interface that can be applied to a

variety of systems. Moreover, we present protocols for experimental setups ranging from custom in vitro assays involving only pure diffusion to in vivo assays involving complex geometries, tortuosity and reaction-kinetics in zebrafish embryos and show how such assays can be used to compose complex FRAP experiments step-by-step. These assays in combination with PyFRAP allow to selectively measure the contribution of factors such as sample geometry and reaction-kinetics to FRAP analysis results and might unravel how global diffusion rates are established. We present in vitro results for fluophore-dextrans ranging over two orders of magnitude in molecular weight and compare the resulting diffusion coefficients with literature values, showing that PyFRAP is able to report diffusion coefficients over multiple time scales. Through inserting polyacrylamide beads into the in vitro assays, we are able to report that tortuosity can decrease global diffusion rates up to twofold for a multitude of macromolecules. We show that tortuosity is similar in vivo and in vitro for recombinant GFP by performing FRAP experiments in living zebrafish embryos. By repeating the same in vivo experiments with secreted GFP, we display that PyFRAP is able to account for underlying kinetics such as production. Finally, we measure the diffusion rate of squint-GFP, a TGF β protein that is crucial for germ-layer patterning, to illustrate that extracellular binding might decrease FRAP diffusion rate estimates up to five-fold.

1 Introduction

Macromolecules control various mechanisms in development and homeostasis. To achieve this, they often need to be transported from their source lo-

cation to their target. For example, the Nodal protein Squint, a key player in the determination of germ-layer formation and a member of the TGF β superfamily is secreted at the marginal zone, however reaches deep into the tissue during zebrafish development (?). Most transport mechanisms are suspected to be regulated by the macromolecule's biophysical properties, such as its stability (**need ref!**) and diffusion rate (**need ref!**). In 1974 both Poo and Cone and Liebman and Entine simultaneously describe a method for measuring diffusion rates of fluorescent macromolecules: During Fluorescence Recovery After Photobleaching (FRAP) assays the fluorescence of molecules in a given region in the sample is bleached by exposure to a strong laser pulse. When the region is sufficiently bleached, one monitors the recovery of the fluorophore intensity inside the region until full recovery. If diffusion happens at a faster time-scale than other kinetics, the observed monitored recovery can then be used to compute the diffusion rate of the fluorescent macromolecule. The computation of diffusion rates is mostly done by fitting curves computed from theoretical solutions to the measured recovery curves (**need ref!**). However, such theoretical solutions rely on a set of assumptions that are not necessarily fulfilled:

Most current analysis methods reduce the FRAP analysis to an either one-dimensional or two-dimensional problem, often ignoring that biological samples have more complex geometries that could play important roles in molecule movement (**need ref!**). For instance, during zebrafish embryogenesis, the germ layers are defined starting at dome stage. Dome stage is characterized by a roughly 15 cell thick layer on top of the embryo's yolk sack. While a FRAP recovery curve of such a sample is dominated by the concentration close to the location of imaging, concentrations close to the yolk will contribute later to the recovery.

Especially in *in vivo* experiments, the investigated fluorescent molecules are generated by injected mRNA and degrade over time. Since fluorescence recovery in such samples often takes up to an hour, the production and degradation can contribute to the observed recovery curve, however are mostly neglected in classical FRAP analysis models.

Moreover, the bleaching process in FRAP experiments is often modeled too ideal: The bleaching is assumed to be equal throughout the bleached

ROI, while the molecules outside of the bleached ROI stay unbleached. However, if the sample is inhomogeneous due for example having both intracellular and extracellular regions or if there is a delay between bleaching and the time point at which the experiment is started, the initial image of the recovery series shows inhomogeneities both inside and outside of the bleached ROI.

In this publication we present PyFRAP. PyFRAP is a versatile Python-based FRAP analysis software that accounts for the previously mentioned drawbacks of current analysis methods: Instead of fitting a curve derived from a theoretical model, PyFRAP simulates the actual FRAP experiment using a finite element scheme and then fits the resulting curve to the experiment data. This allows to tackle all drawbacks at the same time: PyFRAP simulates the FRAP experiment in a three-dimensional geometry, using an interpolation of the initial post-bleaching image as initial conditions, solving the problem of geometry and realistic bleaching conditions. During the fitting process, PyFRAP is able to add both production and degradation to the solution of the simulated FRAP experiment.

To facilitate FRAP analysis, PyFRAP comes with an easy-to-handle graphical user interface (GUI). It allows users even without any computational background to access the whole PyFRAP suite, ranging from different image manipulation methods to improve image analysis to statistical model comparison methods. Moreover, PyFRAP is completely open source, comes with a fully documented API, making it easily customizable to the user's needs and sharing it on Github allows rapid growth of expansion of the PyFRAP software suite through collaborative work. Together with PyFRAP, we present a series of both *in vitro* and *in vivo* FRAP protocols. To benchmark our FRAP analysis, we performed FRAP experiments for fluorophore-dextran ranging from 3kDa dextrans to 500kDa dextrans and recombinant GFP, covering multiple orders of magnitude. Our predicted diffusion rates are overall in agreement with literature values, showing that our *in vitro* assay together with PyFRAP deliver robust diffusion estimates over multiple time-scales. In more realistic setups, the diffusing molecule often has to overcome obstacles such as cells, leading to a tortuous movement. To investigate the influence of such obstacles, we per-

formed similar *in vitro* experiments with polyacrylamide beads mixed into the sample for recombinant GFP and multiple fluophore-dextran. We found that tortuosity can decrease global diffusion rates up to twofold. The transition from *in vitro* to *in vivo* is done by measuring global diffusivity of injected recombinant GFP in living zebrafish embryos at dome stage. Our results indicate that GFP diffusivity *in vivo* are comparable to our *in vitro* results with beads. By injecting GFP mRNA instead of the recombinant protein into the zebrafish embryo, we add production to our system. Due to PyFRAP's ability to incorporate reaction kinetics, we are able to measure the same diffusivity compared to the *in vivo* experiments with recombinant GFP, showing that our software is able to successfully account for simple reaction kinetics without skewing diffusion rate estimates. Extracellular binding is supposed to limit diffusion and accountable for differences in macromolecule diffusivity (?). To test by how much extracellular binding can reduce GFP diffusivity, we injected the Nodal protein Squint-GFP into zebrafish embryos and measured its diffusivity. We are able to reproduce diffusion rates as reported in (?), showing that extracellular binding in this case lowers the diffusion rate fivefold.

The set of *in vitro* and *in vivo* assays presented in this publication in combination with PyFRAP shows how FRAP measurements can be simplified to dissect complicated FRAP measurements to identify the contribution of single factors to the measured global diffusion coefficient.

2 Methods

2.1 PyFRAP

PyFRAP is an open-source Python-based FRAP analysis software that runs on all major operating systems Microsoft Windows, Mac OSX and Linux. PyFRAP comes with an intuitive GUI, a fully documented API allowing quick creation of scripts or modifications of the PyFRAP code, manifold of image processing functions specifically useful for FRAP analysis, arbitrary geometry and ROI definition, a finite element PDE solver that simulates FRAP experiments with various options, statistical tools for results comparison, averaging and model compar-

ison and multiple plotting and I/O functions. A FRAP analysis with PyFRAP can be divided into 4 subtasks:

- Defining experiment conditions,
- running the image analysis,
- simulation the FRAP experiment, and
- fitting the simulation to the analyzed data.

We will now describe these steps more in detail.

2.1.1 Experiment definition

To achieve the best FRAP analysis results, it is important to define the experiment as exactly as possible.

2.1.2 Simulation

3 Results

4 Discussion