SOURCE CODE FOR BIOLOGY AND MEDICINE

SOFTWARE REVIEW Open Access

PREdator: a python based GUI for data analysis, evaluation and fitting

Christoph Wiedemann, Peter Bellstedt and Matthias Görlach*

Abstract

The analysis of a series of experimental data is an essential procedure in virtually every field of research. The information contained in the data is extracted by fitting the experimental data to a mathematical model. The type of the mathematical model (linear, exponential, logarithmic, etc.) reflects the physical laws that underlie the experimental data. Here, we aim to provide a readily accessible, user-friendly python script for data analysis, evaluation and fitting. PREdator is presented at the example of NMR paramagnetic relaxation enhancement analysis.

Keywords: Python3, Matplotlib, Paramagnetic relaxation enhancement, Data fitting

Introduction

In nearly all fields of physical, chemical or biological research it is requiered to convert experimental data into mathematical expressions. Particularly the determination of a "best fit" for a series of data points to a mathematical model is a pivotal and potentially time consuming step in the extraction of results and data evaluation.

Nuclear magnetic resonance (NMR) spectroscopy not only provides structural information at the atomic scale on biological macromolecules but also on their dynamics, and hence, a more complete description of the system under investigation. Furthermore, dynamics parameters may contribute also to the understanding of proteins and their interaction with other proteins, nucleic acids or small ligands. The determination of longitudinal (R_1) or transverse (R_2) relaxation rates of protons in biological macromolecules deliver valuable molecular dynamics information on the system under investigation. For example, this information can be used to determine the interaction interface between individual domains or subunits on the basis of surface accessibility studies in situations where no other NMR parameters, *e.g.* nuclear Overhauser enhancement (NOE) or chemical shift perturbation data, are observable. In such cases, surface accessibility studies can be performed by using of chemically inert paramagnetic probes, *e.g.* paramagnetic metals, oxygen or nitroxides as cosolvents [1]. Protein residues located in the interior of proteins or at the interaction interface are shielded from the paramagnetic agent and experience a weak paramagnetic relaxation enhancement (PRE). In contrast, residues located at the solvent accessible surface experience a strong PRE.

PRE can experimentally be derived from longitudinal (R_1) or transverse (R_2) relaxation rate measurements. A sensitive and reliable measure of transverse PREs can be obtained from cross-peak intensities for the state with and without the paramagnetic cosolvent. Relaxation rates are measured by a series of 2D saturation-recovery spectra (^{1}H, ^{13}C-HMBC or ^{1}H, ^{15}N-CRINEPT [2]), in which the time delay during which relaxation takes place is gradually increased. The experiments are repeated with different concentrations of the paramagnetic agent. To extract the relaxation rates the signal intensities are fitted to $I=I_0(1\text{-}e^{-R_i t})$ where I_0 is the intensity after infinite recovery delay, R_i is the longitudinal or transverse relaxation rate and t is the time. The PRE is calculated and is represented by the slope of the relaxation rate as a function of the concentration of the paramagnetic agent [3-5].

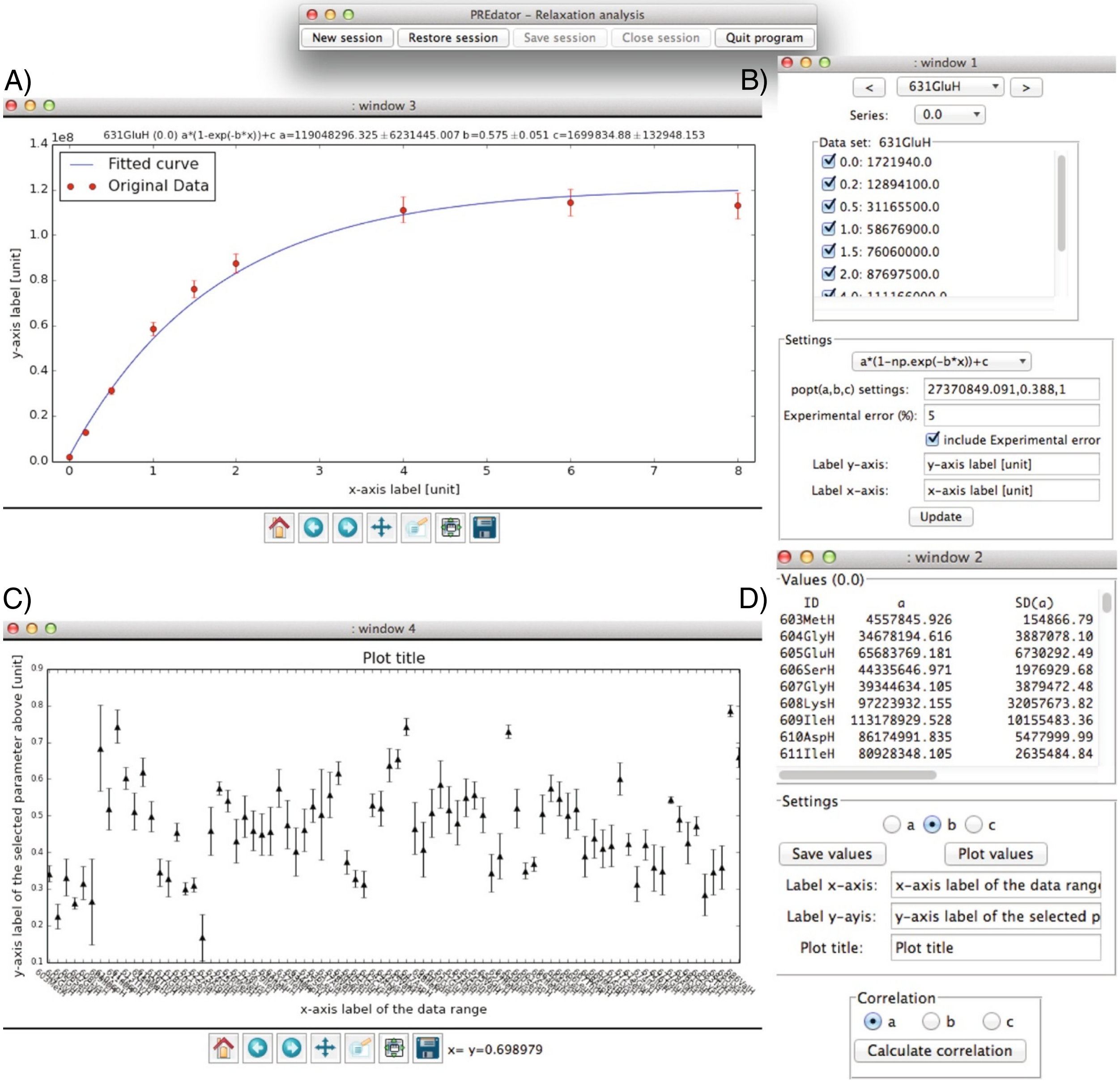
Even though, a variety of tools (*e.g.* MATLAB 8.0 and Statistics Toolbox 8.1 (The MathWorks, Inc., Natick, MA, US), GNU Octave [6] or R [7]) and NMR-software suites (NMRView [8], CCPN [9], ROTDIF [10]) are available for the extraction and fitting of relaxation data, here we provide a straightforward Python3 based application with a graphical user interface not only for the extraction

*Correspondence: mago@fli-leibniz.de
RG Biomolecular NMR Spectroscopy at the Leibniz Institute for Age Research - Fritz Lipmann Institute, Beutenbergstr. 11, 07745 Jena, Germany

of relaxation data but also for the calculation of PREs. However, the script should also be useful for fitting and evaluation of virtually any set of data series.

Implementations and results

PREdator was initially conceived for the analysis of PRE. The application was written in a Mac OS X environment, but it can be run under any operating system for which a Python3 interpreter is available. Python3 and the packages Matplotlib [11], SciPy/NumPy [12] and dill [13] are required to run PREdator.py. Matplotlib [11] is used for data visualisation. All generated plots can be saved as either raster (PNG) or vector format files (PDF or EPS). PREdator also provides the option to save the current session and to restore it later. For data serialization the dill package is implemented [13].



631GluH (0.0) a*(1-exp(-b*x))+c a=119048296.325 ± 6231445.007 b=0.575 ± 0.051 c=1699834.88 ± 132948.153

Series: 0.0

Data set: 631GluH

☑ 0.0: 1721940.0
☑ 0.2: 12894100.0
☑ 0.5: 31165500.0
☑ 1.0: 58676900.0
☑ 1.5: 76060000.0
☑ 2.0: 87697500.0
☑ 4.0: 111166000.0

Settings

a*(1-np.exp(-b*x))+c
popt(a,b,c) settings: 27370849.091,0.388,1
Experimental error (%): 5
☑ include Experimental error
Label y-axis: y-axis label [unit]
Label x-axis: x-axis label [unit]

Values (0.0)

```
   ID          a                 SD(a)
603MetH    4557845.926        154866.79
604GlyH   34678194.616       3887078.10
605GluH   65683769.181       6730292.49
606SerH   44335646.971       1976929.68
607GlyH   39344634.105       3879472.48
608LysH   97223932.155      32057673.82
609IleH  113178929.528      10155483.36
610AspH   86174991.835       5477999.99
611IleH   80928348.105       2635484.84
```

Settings

○ a ● b ○ c
Label x-axis: x-axis label of the data rang
Label y-ayis: y-axis label of the selected p
Plot title: Plot title

Correlation

● a ○ b ○ c

Figure 1 PREdator interface elements. The four interactive analysis windows in PREdator. **A)** Graphical representation of the original data and the fitted curve. **B)** Window for selection or deselection of data points that are considered for fitting, drop down menu for fitting function selection, fields for entering initial fitting parameters and experimental error, and entry fields to enter the axis labels for **A)**. **C)** Graphical summary of one of the selectable fitting parameters over the range of submitted data (*e.g.* amino acid residues). **D)** Summary of the resulting fitting parameters, which can be saved as a text file, is shown in the upper text field. The fitting parameter (a,b or c) selected here is displayed over the data range in **C)**. Entry fields to adjust the title and the axis labels for **C)** are also provided.

The input file has to contain comma separated data (see example files provided with the download package).

In an initial dialogue the user has the opportunity to choose a predefined fitting function from a list or to enter a self-defined fitting function with up to three fitting parameters. The implementation of NumPy allows to create self-defined fitting functions with predefined mathematical expressions (*e.g.* sin, cos or tan).

PREdator provides an initial estimate of the parameters to be fitted. If the user has knowledge of the order of magnitude of the fitting parameters and the experimental error then there is the possibility to enter such initial fitting and error parameters. Data fitting is performed with the curve-fit function implemented in the SciPy-package (modul: scipy.otimize) [12]. For visual inspection the fitted curve and the original data points are shown as graph (Figure 1).

An operating window is provided to re-adjust the fitting function and/or the fitting parameters. Obvious data outliers can be deselected so that they are not considered for fitting. The change of the fitting outcome in the context of the selection and deselection of data points gives a qualitive estimate of fitting robustness. A summary of the fitting results and errors is given in a second window. Fitting errors are provided as one standard deviation errors. The user has the option to save the results to a text file.

For the calculation of the residue-specific PRE, the relaxation rate (R_1 orR_2) for each residue and each concentration of the paramagnetic cosolvent are obtained. The cosolvent concentration dependent relaxation rates for individual residues are subsequently correlated by a second fitting. The slope of the resulting fitted function of this second fitting step delivers the PRE for each individual residue.

PREdator delivers fitting parameters in a first step (*e.g.* R_1 of an individual residue of a protein for different concentrations of the paramagnetic cosolvent). In addition it allows to correlate those fitting parameters, obtained for different conditions, in a second step. The principle of such analysis is not restricted to the evaluation of PREs and is applicable to all kinds of experimental data sets where one type of measurement is repeated under different conditions. Examples include the analysis of fluorescence recovery after photobleaching (FRAP) in a living cell as function of the temperature or the assessment of a DNA-protein interaction under different salt, pH or temperature conditions and to compute properly fitted binding curves. The binding curves in turn can be used to derive the condition-dependent affinity parameter K_d (equilibrium dissociation constant).

Conclusions

In summary, PREdator is a time saving tool for visual inspection, fitting and analysis of series of data points. The application is freely accessible at http://nmr.fli-leibniz.de/nmrsoftware.shtml and can be adapted to user requirements.

Availability and requirements

Project name: PREdator
Project homepage: http://nmr.fli-leibniz.de/nmrsoftware.shtml
Direct Download link: http://nmr.fli-leibniz.de/PREdator/PREdator.zip
Operating systems: Linux, Mac OS X and Windows
Programming language: Python3
Other requirements: Matplotlib, SciPy/ NumPy, dill
License: GNU GPL v3
Any restrictions to use by non-academic users: no licenses are required

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MG is the principal investigator of the project. CW conceived the idea, prepared the NMR samples, recorded and analyzed the NMR spectra. CW and PB programmed the PREdator script. All authors wrote, read and approved the final manuscript.

Acknowledgements

We thank Georg Peiter for technical support and Dr. Peter Hemmerich (FLI Jena) for providing experimental data. CW was supported by the Leibniz Graduated School on Ageing and Age-Related Diseases (LGSA). The FLI is a member of the Science Association 'Gottfried Wilhelm Leibniz' (WGL) and is financially supported by the Federal Government of Germany and the State of Thuringia.

Received: 27 June 2014 Accepted: 17 September 2014
Published: 24 September 2014

References

1. Bertini I, McGreevy KS, Parigi G: *NMR of Biomolecules: Towards Mechanistic Systems Biology*. Weinheim, Germany: John Wiley & Sons; 2012.
2. Riek R, Wider G, Pervushin K, Wüthrich K: **Polarization transfer by cross-correlated relaxation in solution NMR with very large molecules.** *Proc Natl Acad Sci* 1999, **96**(9):4918–4923.
3. Respondek M, Madl T, Göbl C, Golser R, Zangger K: **Mapping the orientation of Helices in Micelle-Bound peptides by paramagnetic relaxation waves.** *J Am Chem Soc* 2007, **129**(16):5228–5234.
4. Madl T, Bermel W, Zangger K: **Use of relaxation enhancements in a paramagnetic environment for the structure determination of proteins using NMR spectroscopy.** *Angewandte Chemie Int Edition* 2009, **48**(44):8259–8262.
5. Madl T, Güttler T, Görlich D, Sattler M: **Structural analysis of large protein complexes using solvent paramagnetic relaxation enhancements.** *Angewandte Chemie Int Edition* 2011, **50**(17):3993–3997.
6. Eaton JW, Bateman D, Hauberg S: *GNU Octave Version 3.0.1 Manual: a High-level Interactive Language for Numerical Computations*: CreateSpace Independent Publishing Platform; 2009. [http://www.gnu.org/software/octave/doc/interpreter]
7. R Core Team: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing; 2014. http://www.R-project.org.
8. Johnson BA, Blevins RA: **NMR View: a computer program for the visualization and analysis of NMR data.** *J Biomolecular NMR* 1994, **4**(5):603–614.
9. Vranken WF, Boucher W, Stevens TJ, Fogh RH, Pajon A, Llinas M, Ulrich EL, Markley JL, Ionides J, Laue ED: **The CCPN data model for NMR**

spectroscopy: development of a software pipeline. *Proteins: Struct Function Bioinformatics* 2005, **59**(4):687–696.

10. Berlin K, Longhini A, Dayie TK, Fushman D: **Deriving quantitative dynamics information for proteins and RNAs using ROTDIF with a graphical user interface.** *J Biomolecular NMR* 2013, **57**(4):333–352.
11. Hunter JD: **Matplotlib: A 2D graphics environment.** *Comput Sci Eng* 2007, **9**(3):0090–0095.
12. Oliphant TE: **Python for scientific computing.** *Comput Sci Eng* 2007, **9**(3):10–20.
13. McKerns MM, Strand L, Sullivan T, Fang A, Aivazis MAG: **Building a framework for predictive science.** In *Proceedings of the 10th Python in Science Conference*. Edited by van der Walt Se, Millman J; 2011:67–78. [http://arxiv.org/pdf/1202.1056]

doi:10.1186/1751-0473-9-21
Cite this article as: Wiedemann *et al.*: PREdator: a python based GUI for data analysis, evaluation and fitting. *Source Code for Biology and Medicine* 2014 **9**:21.