



A Software Library for Monte Carlo-Based Rigid Body Modelling Against Small Angle Scattering Data

Ch. Meesters

published in

From Computational Biophysics to Systems Biology (CBSB08),
Proceedings of the NIC Workshop 2008,
Ulrich H. E. Hansmann, Jan H. Meinke, Sandipan Mohanty,
Walter Nadler, Olav Zimmermann (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **40**, ISBN 978-3-9810843-6-8, pp. 321-324, 2008.

© 2008 by John von Neumann Institute for Computing
Permission to make digital or hard copies of portions of this work for
personal or classroom use is granted provided that the copies are not
made or distributed for profit or commercial advantage and that copies
bear this notice and the full citation on the first page. To copy otherwise
requires prior specific permission by the publisher mentioned above.

<http://www.fz-juelich.de/nic-series/volume40>

A Software Library for Monte Carlo-Based Rigid Body Modelling Against Small Angle Scattering Data

Christian Meesters

Institut für Molekulare Biophysik, Universität Mainz, 55128 Mainz, Germany
E-mail: meesters@uni-mainz.de

Rigid body modelling based on small angle scattering data is a niche in structural biology with rising importance. Here a software package is presented which tries to combine a parameter screen procedure with easy and extensible scripting capabilities. The software is intended as a *proof of concept*-software: The aim is to enable a systematic parameter screen within a reasonable time frame. This goal is reached by a built-in Monte Carlo algorithm with simulated annealing. The software allows for arbitrary symmetries & relations of protein chains. The scripting interface ensures that even “exotic cases” can be handled.

1 Introduction

In order to obtain a structural model from small angle scattering (SAS) two different approaches are followed in general: *ab initio* modelling and rigid body modelling. *Ab initio* modelling is frequently used to get a first impression of the shape of a particle, without need for *a priori* knowledge. Rigid body modelling, on the other hand, is able to provide more details, but requires initial molecular models. Several ways exist to put rigid body modelling into work with SAS data: Some require specific additional restraints, others only rely on the structural information of the model and the scattering³. In contrast to other programs the software presented in this article is fast, does not require restraints obtained by some other method, is not limited to particular point group symmetries, and provides an extensible scripting interface.

The theoretically most precise approach to SAS-wise rigid body modelling is computationally also the most costly one: It is possible to retrace a conformational change, by simply applying (an enormous amount of) possible moves onto the subunits. For each such conformation a check is made how well its theoretical scattering curve fits the experimental data. While this systematic parameter screen provides an accurate estimation for the final conformation – along with uncertainties for each parameter – calculating the theoretical scattering curve many thousands of times is a severe speed bottleneck.

The presented software library divides the task into two steps: First a Monte Carlo-based search in the parameter space is performed, subsequently followed by a conventional parameter screen. In the first step a region of possibilities within the parameter space is found. The final systematic search can then be performed faster, knowing reasonable limits for the movement parameters of the bodies.

2 Software Description

The software is intentionally written as a library in the Python programming language (with some parts in C to gain speed) and not as a stand alone program. This way rapid

prototyping was granted, while potential users are enabled to easily write flexible scripts. Care was taken to have a simple & flexible user interface, while preventing the user to take nonsensical steps. The design strictly follows an object oriented paradigm throughout the construction of this library: At the heart of the library is a base class from which a class for holding atomic models is derived. This class makes use of PDB functionality provided by the biopython project¹. In the future the base class shall provide interfaces for two classes: one holding atomic models and one, yet not released, class for holding electron densities of medium resolution (e. g. from electron microscopy or crystallographic densities of large protein complexes).

3 Modelling Strategy

The programs CRY SOL & CRYSON⁵ serve as plugins for calculating theoretical scattering curves, $I(q)$, for X-rays and neutrons, respectively. In addition users can calculate the distance distribution function, $P(r)$ ^{see 2}. For both cases, $I(q)$ & $P(r)$, a quality factor (χ^2) can be calculated. To fit atomic 3D models to the experimental data, a grid search procedure will screen the parameter space given by rotations and translations for each independent body. The user is asked to specify geometrical restraints and movement limits in advance. The software provides options for defining bodies arbitrarily. It is possible to limit the number of atoms used for calculating the $P(r)$, while a smooth $P(r)$ is warranted. Additional scripting capabilities are provided by using free features of YASARA⁴.

4 The Monte Carlo Algorithm

In order to find reasonable limits to start a systematic parameter screen (see 3) the software provides the option to perform a Monte Carlo based search beforehand (compare figure 1). Provided with SAS data an initial quality factor, χ^2 , is calculated and movements for all individual chains are picked at random. To gain speed the radius of gyration (R_g) for the new model is calculated. A model is immediately rejected, if the R_g falls out of a level of tolerance. Subsequently a new χ^2 is calculated for the new model which is kept ($\chi_{\text{new}}^2 < \chi_{\text{previous}}^2$) or rejected ($\chi_{\text{new}}^2 \geq \chi_{\text{previous}}^2$). In order to prevent falling into a local false minimum, the algorithm might accept a “bad” move with the propability $e^{-\frac{\chi_{\text{previous}}^2 - \chi_{\text{new}}^2}{\sqrt{\chi_{\text{before cooling}}^2}}}$. $\chi_{\text{before cooling}}^2$ is the discrepancy between model and experiment before “cooling down”: The entire system will adjust its own movement limits and search increments, if a certain number of successful steps was reached.

5 Download & Participation

The project homepage can be found at <http://sas-rigid.origo.ethz.ch>. At the time of writing it offers access to the source code and documentation. As Python is an operating system independent language, installing Python (www.python.org) along with some freely available modules (see the documentation in a download) is sufficient to get started. – The project is an open source community project. Any kind of participation is welcome.

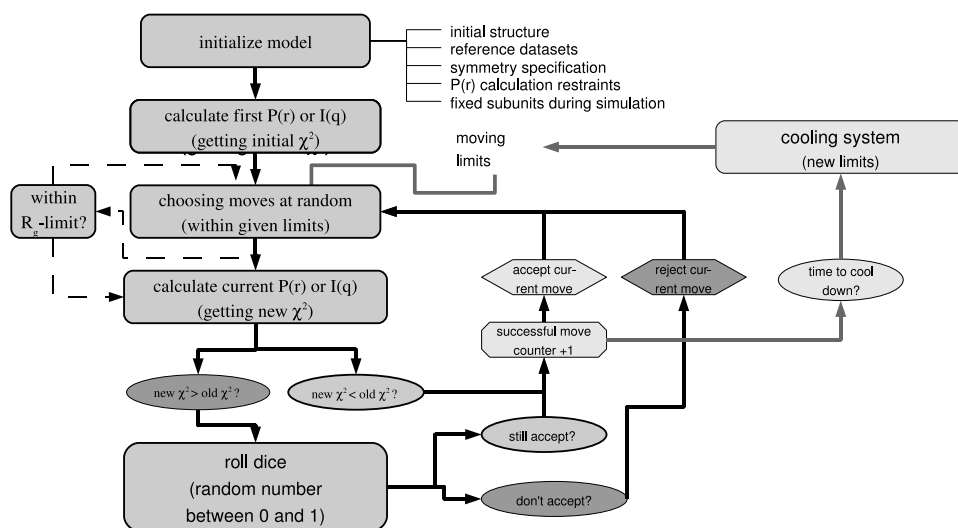


Figure 1. Scheme of the Monte Carlo Algorithm with simulated annealing. (See text.)

Acknowledgments

Work for this project was supported by the Sonderforschungsbereich 490. The author thanks A. Beusch, B. Pairet, and Dr. M. Binder for proofreading.

References

1. T. Hamelryck and B. Manderick, *PDB file parser and structure class implemented in Python*, *Bioinformatics* **19**, 2308–2310, 2003.
2. H. Hartmann and H. Decker, *All hierarchical levels are involved in conformational transitions of the 4 x 6-meric tarantula hemocyanin upon oxygenation*, *Biochimica et Biophysica Acta* **1601(2)**, 132–137, 2002.
3. M.B. Kozin and D. I. Svergun, *Automated matching of high- and low-resolution structural models*, *Journal of Applied Crystallography* **34**, 33–41, 2001.
4. E. Krieger, G. Koraimann, and G. Vriend, *Increasing the precision of comparative models with YASARA NOVA—a self-parameterizing force field*, *Proteins* **47(3)**, 393–402, 2002.
5. D. Svergun, C. Barberato, and M. H. J. Koch, *CRY SOL - a Programm to Evaluate X-ray Solution Scattering of Biological Macromolecules from Atomic Coordinates*, *Journal of Applied Crystallography* **28**, 768–773, 1995.

