COMMUNICATION

OBC www.rsc.org/obc

BioSimGrid: towards a worldwide repository for biomolecular simulations†

Kaihsu Tai,*[a] Stuart Murdock,[b,c] Bing Wu,[a,d] Muan Hong Ng,[c] Steven Johnston,[c] Hans Fangohr,[c] Simon J. Cox,[c] Paul Jeffreys,[d] Jonathan W. Essex[b] and Mark S. P. Sansom*[a]

[a] *Department of Biochemistry, University of Oxford, Rex Richards Building, South Parks Road, Oxford, UK OX1 3QU. E-mail: team@BioSimGrid.org; Fax: +44 1865 275182; Website: http://www.BioSimGrid.org/*

[b] *School of Chemistry, University of Southampton, Highfield, Southampton, UK SO17 1BJ*

[c] *Southampton Regional e-Science Centre, University of Southampton, Building 25, Highfield, Southampton, UK SO17 1BJ*

[d] *Oxford e-Science Centre, University of Oxford, 13 Banbury Road, Oxford, UK OX2 6NN*

Received 26th July 2004, Accepted 9th September 2004

First published as an Advance Article on the web 22nd September 2004

BioSimGrid is a database for biomolecular simulations, or, a 'Protein Data Bank extended in time' for molecular dynamics trajectories. We describe the implementation details: architecture, data schema, deposition, and analysis modules. We encourage the simulation community to explore BioSimGrid and work towards a common trajectory exchange format.

Introduction

Comparative analysis of multiple molecular dynamics (MD) simulations of biomolecules[1,2] should enable us to explore functional patterns of conformational dynamics,[3–5] complementing experimental methods. However, the absence of an accessible database for simulations precludes this and related analyses; the situation is aggravated by the policy of the Protein Data Bank (PDB) not to include 'theoretical' structures. The aim of BioSimGrid[6] is to provide such a database.

Conceptually, a trajectory is a time-series of molecular conformations; that is, it consists of a set of system coordinates for each time point, plus the associated metadata describing the simulation. Indeed, results from Monte Carlo simulations (an ordered series of molecular conformations) may also be stored in BioSimGrid, and we plan extension to also enable storage of data from QM/MM (quantum mechanics/molecular mechanics) simulations. In practice, building and analysing such a database is non-trivial due to the large requirements of processing power and storage capacity. Here we describe our implementation and offer it to the community for evaluation. A detailed description of the current status of the project, including a tutorial manual, is available at http://www.BioSimGrid.org/.

Architecture

BioSimGrid uses a multi-tier client-server architecture.[7] The design decisions, driven by the requirements of the computational chemistry community, are detailed elsewhere.[8] The components involved are shown in Fig. 1.

On the top, two types of user clients are possible in the user interface layer. Any computer capable of running a web browser can access the database by interacting with the BioSimGrid web environment through the hypertext transfer protocol (HTTP). For customized calculations using the data in BioSimGrid, a reasonably powerful workstation with Python[9] installed can interact with the BioSimGrid environment by calling functions in the BioSimGrid application programming interface.

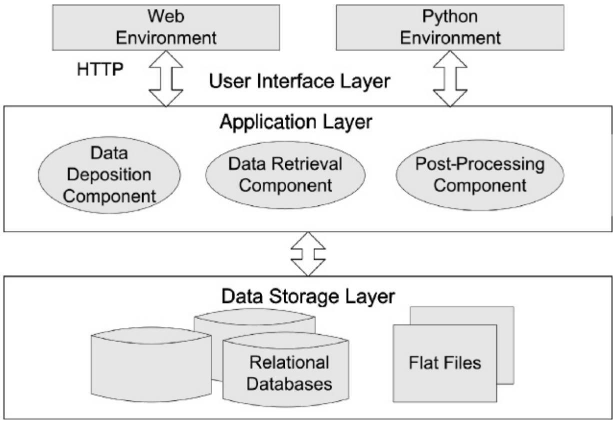

Fig. 1 The architecture of BioSimGrid.

The application layer is in charge of fulfilling the requests coming from the web environment and the Python environment. It provides data deposition, retrieval, and post-processing services. This layer provides abstraction for the access to the data storage layer on the bottom, thus allowing the latter to be heterogeneous.

The database storage layer is in charge of storing and preserving data and fulfilling basic database queries. This layer is heterogeneous to minimize space requirement, and to maximize performance and efficiency: it is a relational database[10] plus structured flat-file subsystems; the evidence supporting this design is presented elsewhere.[8] Any datum here is stored in distributed duplicate (that is, stored in at least 2 sites) for resilience. We plan to enable interrogation and data-fetching from additional sites, using distributed database access. An added benefit of this distributed architecture is the possibility of 'in-house sites' set up by parties for which privacy of data is essential; they can set permissions to allow public access to only those trajectories they would like to share, whilst keeping all other ones private.

Data schema

The data schema (Fig. 2; full version on the website) abstracts the structure of data: this conceptual tool frees us from having at all times to consider the heterogeneous physical storage aspect: the relational database and the structured flat-file storage.

Most of the data (data 'in' the trajectories) are stored in the tables 'trajectory', 'frame', 'coordinate', and 'velocity'. The topology of the biomolecular system is contained in the tables 'chain', 'residue', and 'atom'; this mirrors the conventional PDB hierarchy. The rest of the metadata (data 'about' the trajectories) are in the other tables, with supporting dictionary tables.

† This is one of a number of contributions on the theme of molecular informatics, published to coincide with the RSC Symposium "New Horizons in Molecular Informatics", December 7th 2004, Cambridge UK.

DOI: 10.1039/b411352g

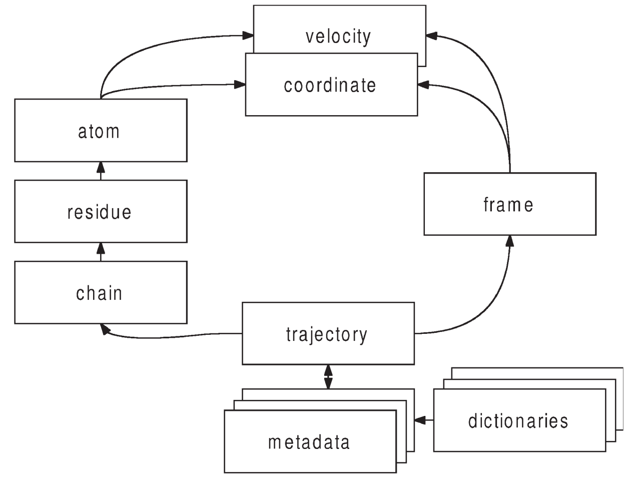

Fig. 2 The abridged data schema for BioSimGrid; the full version is available on the website.

The table ‘trajectory’ brings together the data and the metadata. An entry in this table (a trajectory) may own a certain number of frames. Each frame may own a certain number of coordinates. The ‘frame’ table also stores properties with a time-series nature, such as volume, pressure, and temperature. Each coordinate entry stores the three-dimensional position of an atom in a frame.

In practice, the tables with the most entries, namely ‘coordinate’ and ‘velocity’, are in flat-file format (in distributed duplicate for resilience); the other tables are replicated at all sites in relational database instances.

Deposition

We have developed modules to semi-automate deposition of trajectory files from Gromacs,[11] AMBER,[12] CHARMM,[13] and NAMD;[14] these modules can be adapted for other MD packages.[8] Rather than a totally automated process, some (ideally minimal) intervention by a human curator acting for BioSimGrid is necessary in order to ensure the integrity of data; this procedure mirrors that used by the PDB.[15]

In practice, the user or curator deposits a trajectory by providing the files containing the parameters, the topology, and the coordinates describing the trajectory’s content. For example, to deposit a Gromacs trajectory, the Python script needed is simply:

```
from bioSim.Settings import UserSettings
from bioSim.Deposit.GromacsDeposit import GromacsDeposit
files = {
  ’parameters’: "example.mdp",
  ’topology’: "example.pdb",
  ’coordinates’: ["example1.xtc", "example2.xtc"]
}
mySettings = UserSettings.UserSettings("guest")
GromacsDeposit.GromacsDeposit(mySettings, files)
```

Several long trajectories[16,17] have so far been deposited. We envisage considerable expansion in the near future as trajectories from BioSimGrid consortium members and other colleagues are deposited. One barrier to deposition of trajectories is the variety of trajectory formats and, more importantly, the problems of capturing complete sets of metadata. This suggests a need for a unified simulation data exchange format, analogous to mmCIF.[18]

Analysis modules

There are different ways of performing an analysis on the trajectories for users with specific needs. Browsing the database is possible through the web environment or the Python environment.

The web environment provides functions such as link-out to the PDB, interatomic distance plot for selected atoms, export to PDB file, and export of animation for rendering in VMD[19] or other molecular viewers. Here is a graphical means of selecting different analysis methods to be performed on data which allows investigators with limited computational experience to access results which have already been computed.

Expert users have access to the database and the analysis methods using the Python interpreter and programming language. This allows more experienced investigators to modify standard BioSimGrid analysis for their specific research needs. The BioSimGrid interface also allows authoring of completely new tools for novel analyses; authors may later donate the new modules to the community if they wish. As all coordinates are stored, it is possible to analyse solvent and ion movements[20] in addition to those of the solute. Further, restart data may be retrieved, so trajectory may be extended as more computing resources become available. The language of choice in which to write the analysis tools is Python, as this is currently used in the biosimulation community and it provides a powerful object-orientated scripting language which is relatively easy to learn.

The BioSimGrid interface in the Python environment provides the fundamental analysis tools – including root mean square deviation and fluctuation, average structure, internal angles, interatomic distances, molecular volumes and surface areas, and other geometrical properties. These enable users to perform diverse studies on the trajectories. Results from these tools can be stored in the BioSimGrid environment. Here is a short example analysis script, after loading the necessary BioSimGrid modules and settings:

```
toAnalyse = FCSettings.FCSettings(mySettings, ’2,11-15’)
myFrames = FrameCollection.FrameCollection(toAnalyse)
myRMSD = RMSD.RMSD(myFrames)
myRMSD.createPNG()
```

This computes the root mean square deviation (RMSD) and plots it in the portable network graphics (PNG) format; it is also possible to export to intermediate formats for subsequent data processing.

Assessment of simulation quality

An important issue for any biological database is to measure and indicate to the user the quality of the data. If one considers the PDB, the quality of an X-ray structure is indicated by the resolution of the data and the R_{free} value resulting from refinement. Similarly, for an NMR structure the number of experimental restraints per residue provides a quality indicator.

For biomolecular simulation data, assessing the quality of a simulation is a non-trivial issue. To some extent, defining a set of quality indicators will require input and consensus from the simulation community, and therefore such standards will evolve alongside growth of a simulation database. However, it is possible to define a number of preliminary criteria by which the quality of a simulation may be judged, including:

- ‘Raw’ simulation metadata, such as potential energy and temperature *versus* time;
- C_α RMSD *versus* time, providing a measure of conformational stability or drift (whilst taking into account different levels of conformational drift depending on the nature of the starting structure, *e.g.* NMR or X-ray diffraction[21]);
- Comparison of experimental and simulation-derived crystallographic B-factors for each residue;
- Analysis of overall mean square fluctuations as a function of the duration of the time window over which they were averaged;[22] and
- Radius of gyration of a protein *versus* time.

It should be noted, of course, that the appropriate measures of simulation quality will depend upon the nature of a simulation. For example, a simulation aimed at exploring pathways of protein unfolding would be expected to have a very different RMSD *versus* time profile from a simulation aimed at reproducing the dynamics of a protein within a crystal lattice. Furthermore, as with experimental protein structures, a significant measure of simulation quality is provided by publication in a peer-reviewed journal. Thus, simulation metadata will include bibliographic details.

Application to biomolecular systems

We outline two examples where BioSimGrid is a necessary tool:

The first is a conformational comparison of different trajectories produced over a range of temperatures by different methods. The *Escherichia coli* protein, dihydrofolate reductase, has been simulated *via* a standard MD technique over a range of temperatures to observe how the simulation evolves as the temperature changes. The motion in one of the loops of the protein is vital for the enzyme's catalytic cycle. Similar trajectories were produced with the RDFMD (reversible digitally-filtered molecular dynamics) method[23] and also with a parallel tempering[24] technique. We wish to observe the nature of any differences between the trajectories produced by these different methods, as understanding protein flexibility is an important way of designing inhibitors.

The second is the comparison of two enzyme simulations (Fig. 3): (i) acetylcholinesterase,[16] a key enzyme in the nervous system; and (ii) bacterial outer-membrane phospholipase,[17] a bacterial enzyme involved in pathogenesis. Structural data show that these two enzymes have similar active sites (a triad of amino acid side chains involved in their catalytic mechanisms). The structures of the two proteins are otherwise unrelated. We are analysing simulations to compare the patterns of catalytic side chain dynamics in these two distantly related enzymes, so as to understand the relationship between side chain mobility and catalytic mechanism (Tai *et al.*, manuscript in preparation).

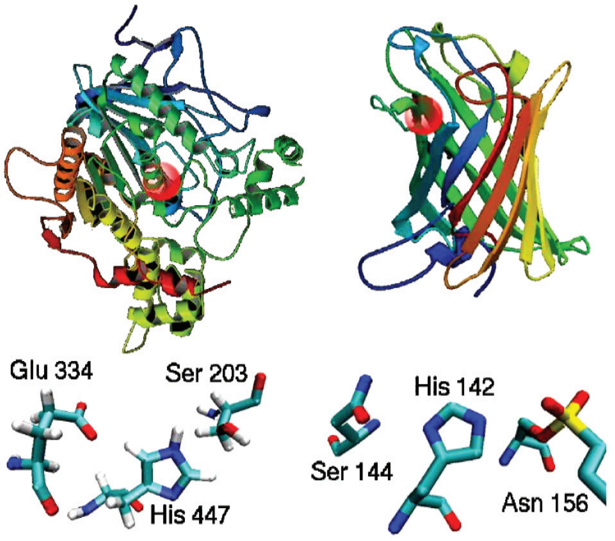

Fig. 3 Comparison of protein simulations contributes to biomedical knowledge. Mouse acetylcholinesterase (left) and bacterial outer-membrane phospholipase (right) are different in structure (top; red spots mark active sites) but similar in their active sites (bottom; with residue type and number) and catalytic function.

Acknowledgements

We thank Oliver Beckstein, Peter Bond, Leo Caves, Matthew Dovey, Charles Laughton, Richard Maurer, David Moss, Adrian Mulholland, Oliver Smart, Kim Henrick and his group for discussions. This project is funded by the Biotechnology and Biological Sciences Research Council, United Kingdom.

Notes and references

1 M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1994.
2 M. Karplus and J. A. McCammon, *Nat. Struct. Biol.*, 2002, **9**, 646–652.
3 A. Pang, Y. Arinaminpathy, M. S. P. Sansom and P. C. Biggin, *FEBS Lett.*, 2003, **550**, 168–174.
4 S. B. Dixit and D. L. Beveridge, Analysis of DNA simulation trajectories using relational database and web based tools, in *226th American Chemical Society National Meeting Abstracts*, American Chemical Society, Washington DC, 2003.
5 M. Abdullah, SimDB – A Grid Software Environment for Molecular Dynamics Simulation and Analysis: Design and User Interface, Master's thesis, University of Houston, 2002.
6 B. Wu, K. Tai, S. Murdock, M. H. Ng, S. Johnston, H. Fangohr, P. Jeffreys, S. Cox, J. Essex and M. S. Sansom, BioSimGrid: a distributed database for biomolecular simulations, in *Proceedings of UK e-Science All Hands Meeting 2003*, ed. S. J. Cox, EPSRC, Swindon, 2003.
7 B. Wu, M. Dovey, M. H. Ng, K. Tai, S. Murdock, H. Fangohr, S. Johnston, P. Jeffreys, S. Cox, J. Essex and M. S. P. Sansom, *J. Digit. Inf. Manag.*, 2004, **2**, 74–78.
8 M. H. Ng, S. Johnston, S. Murdock, B. Wu, K. Tai, H. Fangohr, S. Cox, J. Essex, M. Sansom and P. Jeffreys, Efficient data storage and analysis for generic biomolecular simulation data, in *Proceedings of UK e-Science All Hands Meeting 2004*, EPSRC, Swindon, 2004, pp. 443–450, ISBN 1-904425-21-6.
9 G. van Rossum and F. L. Drake, Jr, *An Introduction to Python*, Network Theory Ltd., Bristol, 2003.
10 C. J. Date, *An introduction to database systems*, Addison-Wesley, Reading, Massachusetts, 2000.
11 E. Lindahl, B. Hess and D. van der Spoel, *J. Mol. Model.*, 2001, **7**, 306–317.
12 D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel and P. Kollman, *Comp. Phys. Commun.*, 1995, **91**, 1–41.
13 B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan and M. Karplus, *J. Comput. Chem.*, 1983, **4**, 187–217.
14 L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan and K. Schulten, *J. Comput. Phys.*, 1999, **151**, 283–312.
15 The PDB Team, The Protein Data Bank, in *Structural bioinformatics*, ed. P. E. Bourne and H. Weissig, Wiley-Liss, Hoboken, New Jersey, 2003.
16 K. Tai, T. Shen, R. H. Henchman, Y. Bourne, P. Marchot and J. A. McCammon, *J. Am. Chem. Soc.*, 2002, **124**, 6153–6161.
17 M. Baaden, C. Meier and M. S. P. Sansom, *J. Mol. Biol.*, 2003, **331**, 177–189.
18 J. D. Westbrook and P. M. D. Fitzgerald, The PDB format, mmCIF formats, and other data formats, in *Structural bioinformatics*, ed. P. E. Bourne and H. Weissig, Wiley Liss, Hoboken, New Jersey, 2003.
19 W. Humphrey, A. Dalke and K. Schulten, *J. Mol. Graphics*, 1996, **14**, 33–38.
20 R. H. Henchman and J. A. McCammon, *Protein Sci.*, 2002, **11**, 2080–2090.
21 H. Fan and A. E. Mark, *Proteins: Struct. Funct. Bioinf.*, 2003, **53**, 111–120.
22 J. D. Faraldo-Gómez, L. R. Forrest, M. Baaden, P. J. Bond, C. Domene, G. Patargias, J. Cuthbertson and M. S. P. Sansom, *Proteins: Struct. Funct. Bioinf.*, 2004 (in press).
23 S. C. Phillips, M. T. Swain, A. P. Wiley, J. W. Essex and C. M. Edge, *J. Phys. Chem. B*, 2003, **107**, 2098–2110.
24 U. H. E. Hansmann, *Chem. Phys. Lett.*, 1997, **281**, 140–150.