Lipidbook: a public repository for force field parameters used in membrane simulations

Jan Domański, Phillip J. Stansfeld, Mark S. P. Sansom, and Oliver Beckstein*

Structural Bioinformatics and Computational Biochemistry Unit, Department of Biochemistry, University of Oxford, Oxford OX1 3QU, UK

*to whom correspondence should be addressed (telephone: +44 (0) 1865 613304; fax: +44 (0)1865 613238; e-mail: oliver.beckstein@bioch.ox.ac.uk)

Citation: Jan Domański, Phillip J. Stansfeld, Mark S. P. Sansom, and Oliver Beckstein. Lipidbook: a public repository for force field parameters used in membrane simulations. *Journal of Membrane Biology* (2010), doi:10.1007/s00232-010-9296-8

Abstract

Lipidbook is a public database for force field parameters with a special emphasis on lipids, detergents, and similar molecules that are of interest when simulating biological membrane systems. It stores parameter files that are supplied by the community. Topologies, parameters, and lipid or whole bilayer structures can be deposited in any format for any simulation code, preferably under a license that promotes "Open Knowledge". A number of mechanisms are implemented to aid a user judge the appropriateness of a given parameter set for their project. For instance, parameter sets are versioned, linked to the primary citation via PubMed identifier and digital object identifier (DOI), and users can publicly comment on deposited parameters. Licensing and hence the conditions for use and dissemination of academically generated data is often unclear. In those cases it is also possible to provide a link instead of uploading a file. A snapshot of the linked file is then archived using the WebCite® service without further involvement of the user or *Lipidbook*, thus ensuring a transparent and permanent history of the parameter set. *Lipidbook* can be accessed freely online at http://lipidbook.bioch.ox.ac.uk. Deposition of data requires online registration.

Keywords: lipids; computer simulations; force field parameters; database; membrane proteins;

Introduction

Membrane proteins constitute about 25% of known genes but over 50% of drug targets as they occupy key roles in cellular transport and signaling pathways (Landry & Gies, 2008). Molecular simulations of membrane proteins in a native-like

bilayer environment provide atomic-scale insights into their function that are not easily obtainable in any other way (Khalili-Araghi et al., 2009; Lindahl & Sansom, 2008; Marrink et al., 2009). Although force field parameters for proteins and water are well established and published with the major simulation software packages, this is not always the case for parameters for the lipids in the membrane. In many cases protein function is known to crucially depend on the identity of lipids and bilayer composition but nevertheless simulations are often carried out with non-native, model lipids because of the difficulty of obtaining reliable parameters for native lipids. Recent distributions of force fields such as GROMOS, Amber, or OPLS do not always include parameters for the lipid molecules of interest even though many have already been parameterized independently. Such parameter sets are dispersed across published papers, mailing list archives, PhD theses, web sites, or can only be obtained by personal communication from the original authors. For the CHARMM simulation code, many lipid parameters are provided in the standard CHARMM force field files (Feller & MacKerell, 2000) and through the *Membrane Builder* of CHARMM-GUI (http://www.charmm-gui.org), a website that automates the setup of protein/membrane CHARMM simulations (Jo et al., 2007; Jo et al., 2009). NAMD users can build CHARMM POPE and POPC bilayers with the *Membrane* plugin(I. Balabin, unpublished, http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/) in VMD (Humphrey et al., 1996). For Amber and CHARMM, the *Glycam Biomolecule Builder* (R.J. Woods et al., unpublished, http://glycam.ccrc.uga.edu/ccrc/biombuilder/biomb_index.jsp) is an online builder for carbohydrates and related molecules and the underlying force field was recently enhanced with lipid and glyco-lipid parameters (Tessier et al., 2008). An expanding database of lipid and small molecule parameters for the GROMOS force field are curated in the *Automated Topology Builder (ATB) and Repository* (A. E. Mark et al., unpublished, http://compbio.chemistry.uq.edu.au/atb/) in formats appropriate for the Gromacs, GROMOS, and CNS simulation packages.

Although there exist databases for membrane protein structures [*Membrane Proteins of Known 3D Structure*, http://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html (White, 2009)], proteins in implicit membranes [*Orientation of Proteins in Membranes (OPM)*, http://opm.phar.umich.edu/ (Lomize et al., 2006)], and coarse grained bilayers [*CGDB*, http://sbcb.bioch.ox.ac.uk/cgdb/ (Scott et al., 2008)], no general database was available to facilitate lipid membrane simulations. Our new database *Lipidbook* provides a central repository for membrane-related force field parameters for all force fields and simulation packages. It provides a general platform for the dissemination of such parameters and supports the user in assessing the parameters' validity and suitability for a given project.

Data in *Lipidbook* is meant to be “open”, in the sense that anyone should be able to access, use, modify, and redistribute the data, as spelt out in the *Open Knowledge Definition* (Open Knowledge Foundation, 2010). Users are encouraged to apply an “open” license to their own work. Special attention is given to treating data in

accordance with any explicit or implicit license provisions while at the same time providing the user unified and transparent access to all data and the data's history.

Methods

Lipidbook consists of a relational database (mySQL, http://www.mysql.com), flat-file storage on disk, and a web frontend based on the symfony framework (http://www.symfony-project.org). The database stores metadata and the location of the force field files, either on disk on the server or as links into the WebCite® archive (Eysenbach & Trudel, 2005) as described below. The frontend allows the user to interact with the database and the file storage, handles user registration and permission checking, and can also initiate WebCite archival requests. The database was initially populated from published parameters, unpublished in-house parameterizations, and user contributions.

A set of parameters is termed a *package* (Fig. 1). It typically contains a topology building block that describes the connectivity and particle types for the molecule of interest and possibly additional parameter files that augment the description of particle types of the published force field. It may also contain a three dimensional structure of a single molecule or, for membrane-forming lipids, a bilayer patch. Molecular entities such as a lipid or a detergent molecule are defined by their name and, where available, their PubChem compound identifier (http://pubchem.ncbi.nlm.nih.gov). Additional metadata are the force field (e.g. "CHARMM27", "GROMOS96 53a6") and the simulation package file format of the parameter files (e.g. "CHARMM", "Gromacs"). Each package also contains a free-form abstract that briefly describes the parameter set and alerts the user to specific features or points towards other online resources. A package is linked to the scientific literature by adding a reference to at least one primary article in which the parameters were published, typically including the digital object identifier (DOI) and the PubMed identifier (http://www.ncbi.nlm.nih.gov/pubmed/). Free form entry is also possible in order to link to publications not indexed on PubMed. Further citations such as to experimental reports or other related parameterization papers can be added.

The "curator", the creator of a package, is responsible for maintaining and updating a parameter set. Alterations are transparent to users because changes are versioned, time-stamped, and assigned a unique SHA1 checksum. After a grace period for editing and polishing a deposition, the current *version* of the package is automatically "locked" and further updates will create a new version in the package (Fig. 1). A version can also be locked manually as soon as it is deemed stable. Changes can be documented in the abstract of the package and in comments that are attached to each version.

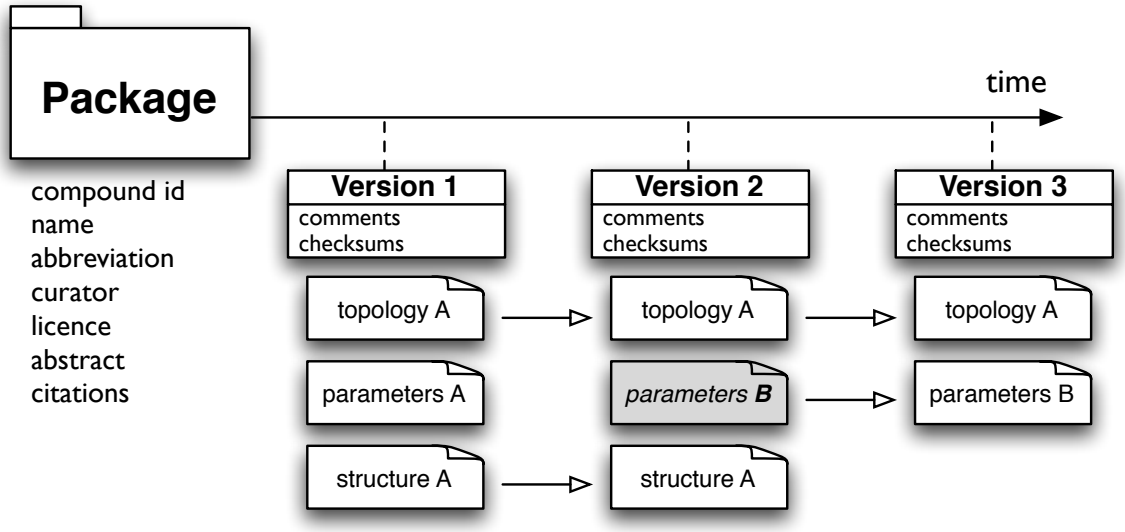

Figure 1. The basic organizational unit in *Lipidbook* is the *package* with its associated metadata. It contains one or multiple *versions*, which in turn reference the locally deposited files or links to files archived through WebCite {Eysenbach, 2005} and comments pertinent to the collection of files. The version number increases with time. Versions in a package are persistent. In the example shown, *parameters A* was updated to *parameters B* in Version 2 and *structure A* was removed from Version 3.

A *package* stores a group of versions in chronological order (Fig. 1), facilitating the tracking of the development of a particular set of files. A *version* contain the data files themselves. Typically, some files change between versions whereas others are simply copied from a previous one. Versions are persistent and never removed, and hence any version in the history of a parameter set is permanently accessible to the user. Locked versions cannot be deleted although the curator can delete a whole package.

In cases where the licensing status of the data is unclear and uploading data could potentially infringe copyright, no data need to be deposited in the database. Instead of uploading, a URI (uniform resource identifier, typically a standard web link) to a file stored elsewhere on the internet can be supplied. In order to guarantee permanent access to the linked file and to have a transparent version history, the linked file is archived through the independent WebCite service (Eysenbach & Trudel, 2005) on www.webcitation.org. In this case, the version in the database contains the unique, time-stamped WebCite URI instead of the file itself. A package can contain both URIs and uploaded files such as equilibrated whole bilayer systems generated from linked parameters.

Usage

Users find packages by either browsing the database or full text searching. Complete versions can be downloaded as a zip archive that contains all files (or links to archived files) and a text file with the relevant primary references and the package license. Registered users can add comments to any version in a package, noting successful use or problems. Users can also subscribe to an RSS feed of newly uploaded or changed packages to keep abreast of development.

Registered users can upload parameter sets through a web-based interface. They first choose the molecule type, force field, and simulation package. It is trivial to add new force field categories or molecules. A license to publish the parameters under must be selected. *Lipidbook* offers licenses that are compatible with the *Open Knowledge Definition* (Open Knowledge Foundation, 2010) which ensures that parameters are accessible and can be modified and redistributed without undue restrictions. After selecting a license, an abstract should be provided that describes the parameter set. Then the actual parameter files are uploaded or URIs to files provided. Files are classified as "topologies" (specific bond and atom-type information for the parameterized molecule), "parameters" (general parameters that supplement published force field files), "structure" (a pdb file of a single molecule), and "bilayer" (a pdb file of whole bilayer patch that can be used as a starting point for a full simulation). Finally, a primary citation must be provided if the license requires attribution to the original authors. When supplied with the PubMed id of an article, the package editor interface auto-completes the citation data. Packages with incomplete citation information and license that requires attribution are visible in the database but are not downloadable until an appropriate reference has been provided. It is also possible to add references to experimental data on the parameterized lipid in order to facilitate validation of the parameters.

In order to update a parameter set, the curator of the package creates a new version in the package. They select the files (or URIs) that are kept unchanged from the last version and upload any new files or provide URIs that replace or supplement previous ones. Changes can be recorded in the abstract of the package and in the comments that are pertinent to the new version. In a similar manner the curator can delete files. It is also possible to edit the meta-data of a package by, for instance, editing the abstract or adding additional citations.

Discussion

Lipidbook is based on the simulation community's philosophy that the user is responsible for their choice of tools and approaches. Our site simplifies obtaining parameter sets for membrane simulations (together with links to the relevant primary literature) by providing a choice of versioned packages of parameters, but it is no substitute for informed use of these parameters.

To encourage contributions, deposition has been designed to be easy. Package curators are typically experts and hence there is no prescription enforced as to which files have to be supplied. Authors of parameter sets are encouraged to curate their own packages to ensure that any user of their work has the most current version of the parameters and the correct reference to cite their work. Although desirable, the parameter author will not always choose to be the *Lipidbook* curator of their own work, for instance, when time and effort have already been invested in their own online distribution infrastructure. In this case any interested user can simply link to the original files on the author's webpage. The WebCite caching and archival mechanism for URI links (Eysenbach & Trudel, 2005) ensures that a user of *Lipidbook* will obtain exactly the same version of the parameters that the curator referenced at the time of deposition. Using WebCite for parameter sets also avoids potential copyright infringement in cases of unclear licensing status (Eysenbach & Trudel, 2005).

A number of features support the user in assessing the validity and suitability of parameter sets such as the direct link to the primary reference, the curator supplied abstract, and the community-supplied comments. Reproducibility of studies is aided by the versioning system that ensures traceable and documented changes. Any registered user can deposit and thus curate data sets. We encourage registration with full names and affiliation (and possibly a link to a home page) so that users can also use the name of the curator as an additional criterion to evaluate a parameter set.

The new *Lipidbook* website (http://lipidbook.bioch.ox.ac.uk) provides a database and storage facility for force field parameter and related files. It serves the membrane simulation community but the principle could be extended to any small molecule parameter sets as used in computational drug development. The *Lipidbook* framework also lends itself to tagging of parameter files with digital object identifiers so that in the future it should become possible to uniquely reference and obtain the input parameters to a simulation via a DOI, similar to the system that is currently used for structures in the Protein Databank (Berman et al., 2000).

Acknowledgement

We thank Benjamin Hall for technical assistance, Kaihsu Tai for the suggestion to use WebCite, and all our colleagues who already contributed packages. This work was supported by grants from the Wellcome Trust, BBSRC, and the EU (EDICT Project, grant number 201924).

References

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. (2000) The Protein Data Bank Nucleic Acids Res. 28:235-242

Eysenbach, G., Trudel, M. (2005) Going, Going, Still There: Using the WebCite Service to Permanently Archive Cited Web Pages. J Med Internet Res 7:e60

Feller, S., MacKerell, A.D.J. (2000) An Improved Empirical Potential Energy Function for Molecular Simulations of Phospholipids. J. Phys. Chem. B 104:7510–7515

Humphrey, W., Dalke, A., Schulten, K. (1996) VMD — Visual Molecular Dynamics. J. Mol. Graph. 14:33–38

Jo, S., Kim, T., Im, W. (2007) Automated Builder and Database of Protein/Membrane Complexes for Molecular Dynamics Simulations. PLoS ONE 2:e880

Jo, S., Lim, J.B., Klauda, J.B., Im, W. (2009) CHARMM-GUI Membrane Builder for Mixed Bilayers and its Application to Yeast Membranes. Biophys. J. 97:50–58

Khalili-Araghi, F., Gumbart, J., Wen, P.-C., Sotomayor, M., Tajkhorshid, E., Schulten, K. (2009) Molecular dynamics simulations of membrane channels and transporters. Curr. Opin. Struct. Biol. 19:128–137

Landry, Y., Gies, J.-P. (2008) Drugs and their molecular targets: an updated overview. Fundam. Clin. Pharmacol. 22:1–18

Lindahl, E., Sansom, M.S.P. (2008) Membrane proteins: molecular dynamics simulations. Curr. Opin. Struct. Biol. 18:425–431

Lomize, M.A., Lomize, A.L., Pogozheva, I.D., Mosberg, H.I. (2006) OPM: orientations of proteins in membranes database. Bioinformatics 22:623–625

Marrink, S.J., de Vries, A.H., Tieleman, D.P. (2009) Lipids on the Move: Computer Simulations of Bilayer Deformations. Biochimica et Biophysica Acta (BBA) - Biomembranes 1788:149–168

Open Knowledge Foundation. 2010. Open Knowledge Definition (OKD). pp. URL:http://www.opendefinition.org/okd/, Archived at http://www.webcitation.org/5rMnDaWbj on 2010-07-20

Scott, K.A., Bond, P.J., Ivetac, A., Chetwynd, A.P., Khalid, S., Sansom, M.S.P. (2008) Coarse-grained MD simulations of membrane protein-bilayer self-assembly. Structure 16:621–630

Tessier, M.B., DeMarco, M.L., Yongye, A.B., Woods, R.J. (2008) Extension of the GLYCAM06 biomolecular force field to lipids, lipid bilayers and glycolipids. Molecular Simulation 34:349–364

White, S.H. (2009) Biophysical dissection of membrane proteins. Nature 459:344–346