OpenStructure: a flexible software framework for computational structural biology

Marco Biasini¹,², Valerio Mariani¹,², Jürgen Haas¹,², Stefan Scheuβer¹,², Andreas D. Schenk³, Torsten Schwede¹,²,∗ and Ansgar Philippsen¹

¹Biozentrum, Universität Basel, Basel, ²SIB Swiss Institute of Bioinformatics, Basel, Switzerland and ³Department of Cell Biology, Harvard Medical School, Boston, MA 02115, USA

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: Developers of new methods in computational structural biology are often hampered in their research by incompatible software tools and non-standardized data formats. To address this problem, we have developed OpenStructure as a modular open source platform to provide a powerful, yet flexible general working environment for structural bioinformatics. OpenStructure consists primarily of a set of libraries written in C++ with a clearly designed application programmer interface. All functionality can be accessed directly in C++ or in a Python layer, meeting both the requirements for high efficiency and ease of use. Powerful selection queries and the notion of entity views to represent these selections greatly facilitate the development and implementation of algorithms on structural data. The modular integration of computational core methods with powerful visualization tools makes OpenStructure an ideal working and development environment. Several applications, such as the latest versions of IPLT and QMean, have been implemented based on OpenStructure—demonstrating its value for the development of next-generation structural biology algorithms.

Availability: Source code licensed under the GNU Lesser general public license and binaries for MacOS X, Linux and Windows are available for download at http://www.openstructure.org.

Contact: torsten.schwede@unibas.ch

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on July 7, 2010; revised on August 6, 2010; accepted on August 14, 2010

1 INTRODUCTION

We introduce OpenStructure, a flexible software framework for computational structural biology, a solid, yet flexible and versatile toolkit for rapid prototyping of new methods as well as their productive implementation. Typically, method development in structural bioinformatics involves combining different independent software tools, and significant effort is devoted to writing code for input/output operations and format conversions between different packages. This culminates when data and algorithms from different domains are to be combined, e.g. protein structures, protein sequence annotation and chemical ligands. Several software tools and frameworks are available today for molecular modeling, e.g. MMTK (Hinsen, 2000), Coot (Emsley et al., 2010) MolIDE (Canutescu and Dunbrack, 2005), Modeller (Eswar et al., 2008), bioinformatics algorithms libraries, e.g. BALL (Kohlbacher and Lenhof, 2000), workflow automation tools, e.g. Biskit (Grunberg et al., 2007) or KNIME (www.knime.org) and visualization e.g. VMD (Humphrey et al., 1996), PyMol (www.pymol.org), DINO (www.dino3d.org), or SwissPdbViewer (Gněx et al., 2009).

OpenStructure is a flexible software framework tailored for computational structural biology, which combines a C++ based library of commonly used functionality with a Python layer and powerful visualization tools. While PyMol and VMD also combine a scripting environment with sophisticated visualization tools, they are primarily geared toward visualization and less on providing a clean application programmer interface (API) that is easy to use and allows for rapid development of new algorithms. OpenStructure is also designed to easily accommodate interfaces to already existing software. This allows for rapid visually enhanced prototyping of new functionality, making OpenStructure an ideal environment for the development of next-generation structural biology algorithms. For example, new versions of the QMean tools for model quality assessment (Benkett et al., 2009a, b) are based on OpenStructure, as well as the structural analysis tools in ProteinModelPortal (Arnold et al., 2009). Further, work is on the way to implement the next generation of the SWISS-MODEL pipeline using the OpenStructure framework (Arnold et al., 2006; Bordoli et al., 2009).

2 IMPLEMENTATION

In OpenStructure, molecular or chemical entities, such as macromolecules, sequences, alignments or electron density maps, are represented as objects, offering a comprehensive set of functions for data manipulation and information querying. Typically, users interact with a high-level Python interface, while ‘power users’ with high computational requirements access the API at the level of C++.

Functionality in OpenStructure is grouped into modules. Each of these modules consists of a computational core as a shared library of C++ code and a set of Python modules built on top of the exported API. Parts of the computational core and the graphical user interface of the Image Processing Library and Toolkit IPLT (Philippsen et al., 2007) have been incorporated into OpenStructure to offer versatile handling of image data with support for various algorithms in one, two and three dimensions. A graphics module for real-time rendering of molecules, density maps and molecular surfaces provides functionalities for data visualization.

© The Author(s) 2010. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.5), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
We would like to thank Andras Aszodi for inspiring discussion during the conception phase of the project, and Pascal Benkert and Tobias Schmidt for critical feedback. OpenStructure uses the Eigen (http://eigen.tuxfamily.org), FFTW (http://www.fftw.org), Boost (http://www.boost.org), Qt4 (http://qt.nokia.com) and PyQt4 (http://www.riverbankcomputing.co.uk).

3 APPLICATION EXAMPLE

Most users will interact with OpenStructure using Python. The code fragment in Supplementary Table S1 illustrates the expressiveness of the OpenStructure API in combining data from different domains. In this example, we compare the sequence conservation of residues in contact with a ligand with the rest of the protein, quantifying the visually derived hypothesis that the binding-site residues of the SH2 domain are more conserved than the rest. This is achieved by mapping of a conservation score derived from a multiple sequence alignment of various SH2 domains (’sh2.aln’) onto a representative structure (PDB: 3IMJ) (DeLorbe et al., 2009) and identifying residues in direct contact with the ligand. Figure 1 shows the results displayed in the DNG (’DINODeepView Next Generation’) graphical user interface, using the conservation score to color a molecular surface representation.

OpenStructure distribution contains several scripting examples to introduce new users to the functionalities and usage style of the tool kit, such as scripts to animate molecular dynamics trajectories, calculate electron density maps from atomistic structures or rank short peptide fragments according to their correlation with electron density. Exhaustive documentation and tutorials are provided on the web site. Mailing lists for OpenStructure users and developers provide a forum to ask questions, report problems or suggest new developments.

ACKNOWLEDGEMENT

We would like to thank Andras Aszodi for inspiring discussion during the conception phase of the project, and Pascal Benkert and Tobias Schmidt for critical feedback. OpenStructure uses Eigen (http://eigen.tuxfamily.org), FFTW (http://www.fftw.org), Boost (http://www.boost.org), Qt4 (http://qt.nokia.com) and PyQt4 (http://www.riverbankcomputing.co.uk).

Funding: Development of OpenStructure was funded by the SIB Swiss Institute of Bioinformatics and the Biozentrum University of Basel. Implementation of the structure comparison for the Nature PSI SIKB Protein Model Portal based on OpenStructure was supported by the National Institutes of Health as a sub-grant with Rutgers University, under Prime Agreement Award Number: 3U54GM074958-05S2.

Conflict of Interest: none declared.

REFERENCES

