Structural bioinformatics

The Victor C++ library for protein representation and advanced manipulation

Layla Hirsh1,2, Damiano Piovesan1, Manuel Giollo1,3, Carlo Ferrari3 and Silvio C. E. Tosatto1*

1Department of Biomedical Sciences, University of Padua, Viale G. Colombo 3, 35131 Padova, Italy, 2Department of Engineering, Pontificia Universidad Católica del Perú, San Miguel, 32 Lima, Perú and 3Department of Information Engineering, University of Padua, Via Gradenigo 6, 35121 Padova, Italy

*To whom correspondence should be addressed.
Associate Editor: Anna Tramontano

Received on August 19, 2014; revised on November 12, 2014; accepted on November 15, 2014

Abstract

Motivation: Protein sequence and structure representation and manipulation require dedicated software libraries to support methods of increasing complexity. Here, we describe the VIrtual Constrution TOol for pRoteins (Victor) C++ library, an open source platform dedicated to enabling inexperienced users to develop advanced tools and gathering contributions from the community. The provided application examples cover statistical energy potentials, profile–profile sequence alignments and ab initio loop modeling. Victor was used over the last 15 years in several publications and optimized for efficiency. It is provided as a GitHub repository with source files and unit tests, plus extensive online documentation, including a Wiki with help files and tutorials, examples and Doxygen documentation.

Availability and implementation: The C++ library and online documentation, distributed under a GPL license are available from URL: http://protein.bio.unipd.it/victor/.

Contact: silvio.tosatto@unipd.it

1 Introduction

Structural bioinformatics methods require valid software libraries to represent and manipulate proteins efficiently. A number of widely used tools have been developed over the years to visualize proteins, e.g. Chimera (Huang et al., 2014), Swiss-PdbViewer (Guex et al., 2009), MolIDE (Canutescu and Dunbrack, 2005) and VMD (Humphrey et al., 1996) to name a few. Software libraries to manipulate proteins efficiently provide basic data representation and more advanced functionality with a different focus each. ESBTL (Loriot et al., 2010) is mainly a Protein Data Bank (PDB) file parser. Biskit (Grünberg et al., 2007) additionally provides functionality for analysis of molecular dynamics simulations, while PTools (Saladin et al., 2009) focuses on molecular docking. OpenStructure (Biasini et al., 2010) places more attention on structure visualization and energy calculation. The latter is also supported by MSL (Kulp et al., 2012) and Tinker (Shi et al., 2013), while BALL (Hildebrandt et al., 2010) in addition provides many advanced optimization algorithms.

Finally, StrBioLib (Chandonia, 2007) extracts sequence information from the protein structure and can be used as an interface to several available third-party tools.

The critical assessment of techniques for protein structure prediction (CASP) series of experiments (Moult et al., 2014) demonstrates that structure prediction is increasingly becoming an engineering problem, where sophisticated methods have to be combined into extensive pipelines to provide state-of-the-art results (Khoury et al., 2014). This has raised the barrier for entry into the field to a point where little new developments are possible, considering that most software libraries used in CASP are proprietary and not available as open source. Here, we propose the open-source VIrtual Constrution TOol for pRoteins (Victor) C++ library as a way to mitigate this problem. Victor is both an efficiently designed C++ library, able to manipulate protein structures with minimal computing time, and a collection of advanced components for protein sequence and structure manipulation. In particular, Victor...
provides three sample applications: profile–profile sequence alignments (Wang and Dunbrack, 2004), statistical potentials (Tosatto, 2005) and loop modelling (Tosatto et al., 2002). Each of these three applications has been extensively described in the literature and is beyond the scope of this article. To the best of our knowledge, neither is available as an open-source C++ library yet. Profile–profile sequence alignments, in particular, have been widely used to improve target-template alignment in CASP (Kryshtafovych et al., 2014). Victor is composed of >60 000 lines of code and still expanding as it is used in the main author’s teaching. It was developed in-house over the last 15 years with the contribution of tens of developers and has reached a high level of maturity. Victor is released to provide a platform for contributions from the interested community. It provides extensive online material in the form of a Wiki with help files, tutorials, Doxygen documentation and a list of applications built using Victor can be accessed from the URL: http://protein.bio.unipd.it/victor/. The actual GitHub repository with C++ source files, a precompiled Ubuntu 64-bit version and unit tests are available from URL: https://github.com/BioComputingUP/Victor.

2 Core library

The Victor C++ library currently contains two components for data representation and manipulation in separate directories: tools and Biopool. Tools provide basic manipulation methods, e.g. vector coordinates and file I/O. The core of the library is provided by the Biopool module, which defines all relevant data structures and algorithms to represent protein structures and manipulate them at a higher level of abstraction. The core data structures were carefully developed using design patterns (Gamma et al., 1995), to provide an elegant and simple, yet powerful set of C++ classes. To allow the simple manipulation of protein structure through the more intuitive torsion angles, automating low-level geometric transformations, atom positions are coded both explicitly in 3D coordinates and as a position relative to the previous atom on a graph structure. This ensures consistency in the structure, while allowing the programmer to change the protein conformation rotating a torsion angle with a single line of code. Computational efficiency is guaranteed by updating the corresponding Cartesian coordinates only when necessary. All low-level geometrical transformations remain transparent to the user. Biopool is able to read properly all existing PDB files. Additional tools are also provided, such as protein secondary structure automatic assignment with additional terms for sequence to structure fit (Madhusudhan et al., 2005) and advanced weighting schemes such as PSIC (Sunyaev et al., 1999). Alignment parameters have been extensively benchmarked and the default parameters are optimized for performance.

Last but not least, the Lobo directory contains an application of ab initio loop modeling using a fast divide and conquer algorithm (Tosatto et al., 2002). This makes extensive use of the functions to construct novel amino acids and manipulate the protein structure locally, providing sample code for more complex structural manipulations. It can easily be extended for ab initio structure prediction in combination with statistical potentials as target function.

4 Conclusions

The Victor library is an open source project devoted to the structural bioinformatics community. It provides a unique combination of methods for sequence and structure manipulation. Expansion is ongoing both through in-house development, as it is the basis for several more recent publications [e.g. RING (Martin et al., 2011) and NeEMO (Giolo et al., 2014)], and as part of the author’s teaching activities, which include software development projects for students. We hope that the Victor library will contribute towards an easier development of advanced methods for structural bioinformatics.

Acknowledgements

To the Francesco Lovo, Enrico Negri and several students for contributing to the Victor project over the years as well as to members of the BioComputing UP lab for insightful discussions.

Funding

This project was funded by FIRB Futuro in Ricerca grant RBFR08ZSWY and University of Padua grant CPDR123473 to S.T.

Conflict of interest: none declared.

References


Gamma, E. et al. (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, USA.


