BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 3 2006, pages 365–366
doi:10.1093/bioinformatics/bti818

Structural bioinformatics

BALLView: a tool for research and education in molecular modeling

Andreas Moll[1,*], Andreas Hildebrandt[1], Hans-Peter Lenhof[1,†] and Oliver Kohlbacher[2,†]

[1]Center for Bioinformatics, Saarland University, Saarbrücken, Germany and [2]Center for Bioinformatics Tübingen (ZBIT), Eberhard Karls University Tübingen, Germany

Received on November 2, 2005; revised on December 1, 2005; accepted on December 2, 2005
Advance Access publication December 6, 2005
Associate Editor: Anna Tramontano

ABSTRACT

Summary: We present BALLView, a molecular viewer and modeling tool. It combines state-of-the-art visualization capabilities with powerful modeling functionality including implementations of force field methods and continuum electrostatics models. BALLView is a versatile and extensible tool for research in structural bioinformatics and molecular modeling. Furthermore, the convenient and intuitive graphical user interface offers novice users direct access to the full functionality, rendering it ideal for teaching. Through an interface to the object-oriented scripting language Python it is easily extensible.

Availability: BALLView is an open source software and runs on all major platforms (Windows, MacOS X, Linux and most Unix flavors). It is available free of charge under the GNU Public License at http://www.ballview.org

Contact: amoll@bioinf.uni-saarland.de

1 INTRODUCTION

Over the last years, a wide variety of freely available tools for molecular visualization and modeling has been developed [e.g. MESHI (Kalisman *et al*., 2005), PyMOL (Delano, 2002), RasMol (Sayle and Milner-White, 1995), Swiss-PdbViewer (Guex and Peitsch, 1997), VMD (Humphrey *et al*., 1996) and WebLabViewer (Discovery Studio; available at: http://www.accelrys.com/products/dstudio/index.html). Such molecular modeling tools face the challenge of providing a rich functionality while at the same time offering an intuitive interface to their capabilities. This is particularly important if the tool is aimed at inexperienced users as well as at experts. For the latter group it should also be easily possible to extend and adapt such a program to the user's needs. Most of the existing tools focus on the functionality, often at the cost of useability. In contrast, our molecular viewer and modeling tool BALLView tries to bridge this gap by offering rich functionality via a userfriendly adaptable interface equally suited for novice users and experts. Its development has thus been guided by the abovementioned design goals. BALLView is based on the Biochemical Algorithms Library (BALL) (Kohlbacher and Lenhof, 2000), an object-oriented framework for rapid application development in structural bioinformatics. Therefore, BALLView shares BALL's molecular modeling functionality, its extensibility and scripting capabilities. BALLView's state-of-the-art visualization techniques, its rich molecular modeling capabilities and the convenient graphical user interface (GUI) make it a powerful tool for research as well as for teaching structural bioinformatics.

2 DESCRIPTION

BALLView was designed with two main goals in mind: (1) user-friendliness and (2) rich functionality. The latter is provided through our molecular modeling framework BALL (Kohlbacher and Lenhof, 2000). To achieve the user-friendliness, we designed a intuitive, flexible and adaptable GUI, which was built on top of QT (http://www.trolltech.com), an object-oriented GUI library. We will now briefly review some of BALLView's features: for a more detailed description, we refer to the online documentation and tutorial.

Reading molecular structures is a prerequisite to almost all visualization tasks. BALLView reads most of the common molecular structure formats (PDB, MOL, MOL2 and SD) and can export structures to most of these formats as well. Furthermore, the GUI allows to download structures directly from the PDB (Berman *et al*., 2000). Structures can than be visualized with all standard graphical models and coloring methods, which can also be freely combined. This allows the user to visualize complex molecular scenes, e.g. a molecule's charge along with its structure. Since all models can be rendered in four different detail levels it is possible to visualize huge molecules even on slower machines (e.g. laptop computers). High-quality figures are easily created by exporting the models to a ray-tracer (POVRay).

In contrast to most other molecular visualization tools, BALLView also provides access to a rich molecular modeling functionality implemented in the BALL framework. The built-in AMBER and CHARMM force fields can be used for energy minimizations (using steepest descent or conjugate gradient minimizers) or molecular dynamics simulations. The resulting trajectories can be exported and visualized interactively. Additionally, the user can create movies from these trajectories. Molecular mechanics calculations are delegated to a separate application thread: thus, the user interface remains useable while the computation is executed in the background. This allows for interactive molecular dynamics simulations in real time. Furthermore, BALLView provides a convenient way to calculate electrostatic potentials similar to DelPhi (Nicholls and Honig, 1990) using a native implementation of a finite-difference Poisson–Boltzmann (FDPB) solver. The

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, H.L. and O.K. contributed equally to this work.

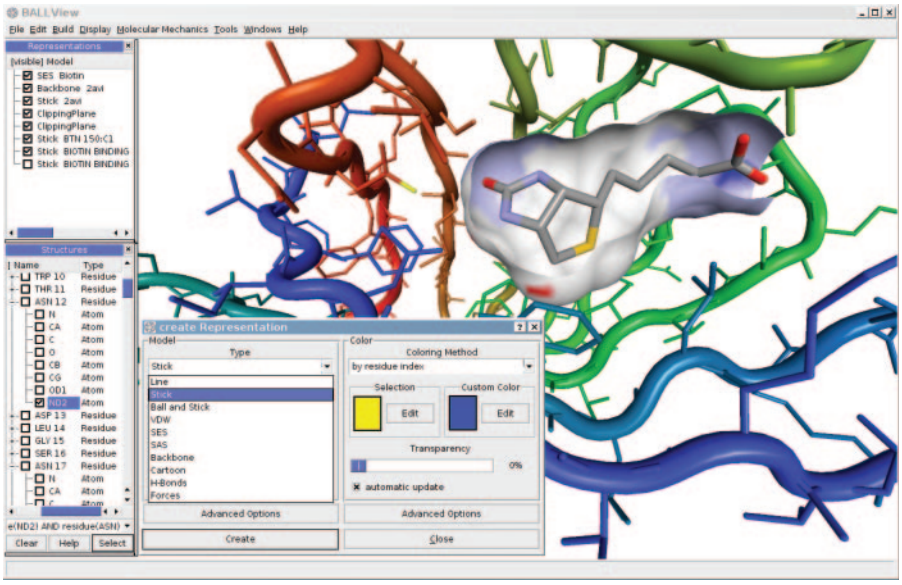

Fig. 1. BALLView's user interface consists of several independent subwindows that can be freely arranged, resized, and completely hidden. The structure widget, here at the lower left, shows all structures currently loaded in a hierarchical fashion. The structures, or parts thereof, can be visualized with arbitrary representations (shown in the subwindow in the upper left corner). The figure shows the binding pocket of avidin with bound biotin (PDB ID 2AVI). The binding pocket's surface is colored by electrostatic potential (calculated using BALLView's integrated Poisson–Boltzmann solver). The binding pocket has been isolated by restricting the solvent-excluded surface of avidin to the region close to the ligand and by the use of two additional clipping planes.

resulting potentials can be visualized by coloring molecular surfaces or as isocontour surfaces similar to GRASP (Nicholls *et al*., 1991). All these and further modeling features are available through a convenient user interface (see screenshot in Fig. 1).

Since BALLView is also intended for teaching, we have made sure that it is easily useable by novice users: a demo and a built-in tutorial ease the first steps with BALLView and context-dependent help texts provide information for the whole interface. Furthermore, BALLView has detailed documentation in HTML format available and the project's website offers support with forums and mailing lists. For the past three years, we have taught a variety of graduate and undergraduate courses in structural biology and bioinformatics using BALLView with huge success.

To allow rapid prototyping, BALLView provides an interface to Python, a powerful, nevertheless easy to read and easy to learn, object-oriented scripting language. This also enables inexperienced users to expand BALLView's functionality or automate frequently occuring tasks. The object-oriented design of the underlying libraries and the application itself simplify the extension tremendously.

Currently, we are adding more modeling features, in particular functionality for editing molecules, aligning structures and detection of binding pockets. Furthermore, we are implementing the Merck Molecular Force Field (MMFF94; Halgren, 1996) and an interface for protein–protein and protein–ligand docking, which will be included in the next major BALLView release.

3 AVAILABILITY

BALLView is an open source software available under the GPL. The BALLView project page at http://www.ballview.org provides both, the source code and precompiled binaries and installers for some of the platforms (Windows and Mac OS X). Compilation on all other platforms (Linux, Solaris, IRIX and most Unix flavors) is simple even for inexperienced users.

ACKNOWLEDGEMENTS

This work was supported in parts by Deutsche Forschungsgemeinschaft grants BIZ 1/1-3, BIZ 4/1-1 and LE 952/2-3.

Conflict of Interest: none declared.

REFERENCES

Berman,H.M. *et al*. (2000) The Protein Data Bank. *Nucleic Acids Res*., **28**, 235–242.

DeLano,W.L. (2002) The PyMOL molecular graphics system. Delano Scientific, San Carlos, CA.

Guex,N. and Peitsch,M.C. (1997) SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis*, **18**, 2714–2723.

Halgren,T.A. (1996) Merck molecular force field: I. basis, form, scope, parameterization and performance of MMFF94. *J. Comput. Chem*., **17**, 490–519.

Humphrey,W. *et al*. (1996) VMD—Visual Molecular Dynamics. *J. Mol. Graph*., **14**, 33–38.

Kalisman,N. *et al*. (2005) MESHI: a new library of Java classes for molecular modeling. *Bioinformatics*, **21**, 3931–3932.

Kohlbacher,O. and Lenhof,H.P. (2000) BALL—rapid software prototyping in computational molecular biology. *Bioinformatics*, **16**, 815–824.

Nicholls,A. and Honig,B. (1990) A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson–Boltzmann equation. *J. Comput. Chem*., **12**, 435–445.

Nicholls,A. *et al*. (1991) Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins*, **11**, 281–296.

Sayle,R. and Milner-White,E.J. (1995) RasMol: biomolecular graphics for all. *Trends Biochem. Sci*., **20**, 374.