BIOINFORMATICS APPLICATIONS NOTE Vol. 24 no. 4 2008, pages 584–585
doi:10.1093/bioinformatics/btm627

Structural bioinformatics

Utility library for structural bioinformatics

Dominik Gront* and Andrzej Kolinski

University of Warsaw, Faculty of Chemistry, Pasteura 1 02-093 Warsaw, Poland

Received and revised on November 20, 2007; accepted on December 15, 2007

Advance Access publication January 28, 2008

Associate Editor: Anna Tramontano

ABSTRACT

Summary: In this Note we present a new software library for structural bioinformatics. The library contains programs, computing sequence- and profile-based alignments and a variety of structural calculations with user-friendly handling of various data formats. The software organization is very flexible. Algorithms are written in Java language and may be used by Java programs. Moreover the modules can be accessed from Jython (Python scripting language implemented in Java) scripts. Finally, the new version of BioShell delivers several utility programs that can do typical bioinformatics task from a command-line level.

Availability The software is available for download free of charge from its website: http://bioshell.chem.uw.edu.pl. This website provides also numerous examples, code snippets and API documentation.

Contact: dgront@chem.uw.edu.pl

1 INTRODUCTION

BioShell (Gront and Kolinski, 2006), a suite of programs published a few years ago has been designed as an extension of a Unix shell with commands related to common bioinformatics tasks. After some time of development it became obvious that a set of independent programs is not the most efficient way for performing a variety of computational tasks. Adding new programs and extending their list of command-line options was not enough to meet the needs of all users, both from inside and outside of our laboratory. Therefore the idea of BioShell has been radically modified. The package has been rewritten in a highly modular and object-oriented fashion. After two years of its development, BioShell moves toward a general biomodeling scripting language.

2 BIOSHELL OVERVIEW

In this article we present a large library of modules written in JAVA language. The library is not dedicated solely for JAVA programmers. The new BioShell functionality follows a novel approach to toolkit construction and may be accessed in three ways:

- as a set of command line tools. This has not been changed in respect to the previous BioShell distribution, besides a few new commands (executable programs) that were added recently. From this point of view BioShell resembles such command line packages as EMBOSS (Rice *et al.*, 2000).
- as a library of modules for Python language. Java classes may be directly called from Python language providing the Python interpreter itself has been implemented in JAVA, such as Jython.[1] This way of BioShell usage follows such Bio* projects as BioPython (Hamelryck and Manderick, 2003) or BioPerl (Stajich *et al.*, 2002). When compared to these two scripting packages, BioShell offers a wider range of possible applications, focusing on various aspects of structure analysis rather than solely on the sequence-based bioinformatics. Scripting with BioShell is intended to be the main way to access its modules.
- finally the **jbcl** (**J**ava **B**io**C**omputing **L**ibrary) that holds over 80% of BioShell code can be directly used to develop JAVA programs. From this side BioShell resembles such JAVA-oriented libraries as BioJAVA (Pocock *et al.*, 2000).

The outlined above software architecture is very helpful for those learning BioShell. Some part of daily work may be done by command-line tools without any programming. Larger projects usually involve preparation of a script. Finally, when the investigated idea has been proved and a script found to be useful, that script may be compiled by a `jythonc` utility from Jython package into a Java bytecode (class file).

BioShell is designed to be easy to learn and to use. In particular, the authors devoted a lot of attention to avoiding an explosion of different kinds of objects, data structures, etc. BioShell methods return a basic data type (such as double integers or String) wherever it is possible. Therefore the number of objects that a user must learn is highly limited. BioShell by itself does not utilizes any external packages or libraries. The only components that must be installed to use BioShell are Java Runtime Environment and Jython interpreter.

2.1 BioShell modules

The package integrates some of our previous projects: T-Pile: (Gront and Kolinski, 2007), BBQ (Gront *et al.*, 2007) and HCPM (Gront and Kolinski, 2005). There also many new

*To whom correspondence should be addressed.

[1]http://www.jython.org

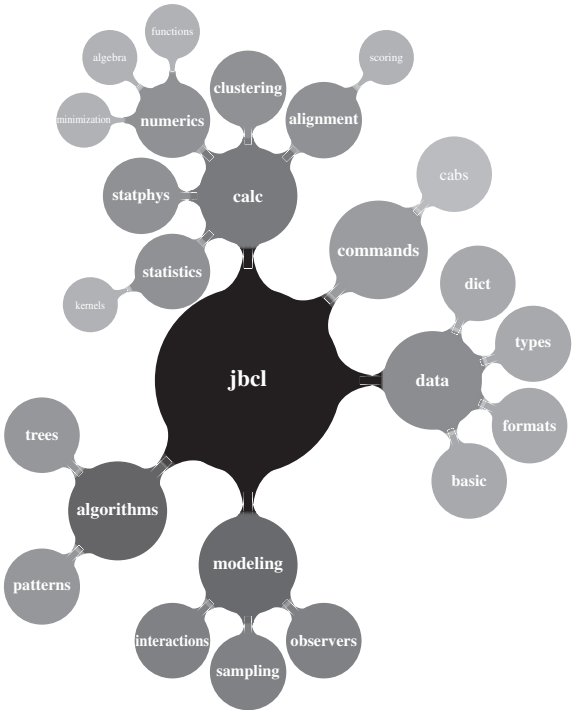

Fig. 1. Graphic representation of jbcl utility library presents the components of BioShell package: **calc** module for calculating alignments and structural properties, contains also various numerical and statistical utilities; **commands** and **algorithms** are used by BioShell command-line applications; **data** modules provide core objects, dictionaries and file format parsers; finally **modeling** contains tools for Monte Carlo sampling. Each branch level corresponds to a level of the modules hierarchy. For example, a PDB reader class may be found in **jbcl.data.formats** package; NormalKernel used in kernel density estimation is located in **jbcl.calc.statistics.kernels**.

modules. Currently the source code counts almost 40.000 lines of Java code in 260 classes. Therefore we must limit ourselves to list only the most important (and probably the most frequently used) packages:

jbcl.data.dict- dictionaries, holding various constants, facts etc, for example formulas of ligands common in PDB files, Van der Waals atomic radii or standard geometry of amino acid residues (bond lengths and planar angles).

jbcl.data.formats- parsers for most popular file formats, such as PDB, FASTA, DSSP, PIR, MOL2 and others.

jbcl.data.types- classes that represent objects typical for bioinformatics: proteins, residues, sequence profiles etc.

jbcl.calc.alignment- classes in this package calculate sequence alignments of various flavors.

jbcl.calc.structural- classes that calculate structural properties, such as protein–protein similarity (crmsd, drsmd, GDT-TS, TM-score), torsion and planar angles and interatomic distances.

[2]http://www.rcsb.org/

2.2 BioShell examples

Because of the limited space in Bioinformatics Application Notes, we cannot bring here any detailed example. The project's website provides four very detailed tutorials:

(1) Parsing PDB (Berman *et al.*, 2000) files. Our PDB parser reads both *.ent and *.ent.gz files. It is also possible to download a protein directly from PDB website.[2] On average it takes ~1 sec to read a protein from a gzipped file. BioShell employs also JAVA serialization technique which gives additional speed-up. When it is applied to PDB I/O operations, it takes ~0.2 sec. to read a PDB entry,

(2) Support for reduced-space modeling,

(3) Calculating various structural properties,

(4) Various kinds of sequence-based alignments, from the very basic to 1D threading.

Moreover, jbcl API documentations contains numerous code snippets and example scripts with many more to come.

3 SUMMARY

In this contribution we present a software package for bioinformatics calculations that can be used as a set of standalone applications, JAVA utility library or as a set of modules for Jython. In contrast to other existing libraries such as BioPython and BioJava, BioShell offers higher flexibility and wider scope of functionality, ranging from sequence-based to structure-based bioinformatics. The current JAVA implementation of BioShell (the previous version has been written in C++) makes the package platform independent. It has been successfully tested on Linux, MacOS, Windows XP and Vista systems. We hope that BioShell modules will be a valuable addition to the publicly available biocomputing software.

Conflict of Interest: none declared.

REFERENCES

Berman,H.M. *et al.* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Gront,D. and Kolinski,A. (2005) HCPM – program for hierarchical clustering of protein models. *Bioinformatics*, **21**, 3179–3180.

Gront,D. and Kolinski,A. (2006) Bioshell – a package of tools for structural biology computations. *Bioinformatics*, **22**, 621–622.

Gront,D. and Kolinski,A. (2007) T-pile- a package for thermodynamic calculations for biomolecules. *Bioinformatics*, **23**, 1840–1842.

Gront,D. *et al.* (2007) Backbone building from quadrilaterals: A fast and accurate algorithm for protein backbone reconstruction from alpha carbon coordinates. *J. Comput. Chem.*, **28**, 1593–1597.

Hamelryck,T. and Manderick,B. (2003) Pdb file parser and structure class implemented in python. *Bioinformatics*, **19**, 2308–2310.

Pocock,M. *et al.* (2000) BioJava: open source components for bioinformatics. *ACM SIGBIO Newsletter*, **20**, 10–12.

Rice,P. *et al.* (2000) Emboss: the european molecular biology open software suite. *Trends Genet.*, **16**, 276–277.

Stajich,J.E. *et al.* (2002) The bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.