BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 5 2006, pages 621–622
doi:10.1093/bioinformatics/btk037

Structural Bioinformatics

BioShell—a package of tools for structural biology computations

Dominik Gront* and Andrzej Kolinski

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

Received on November 14, 2005; revised and accepted on January 2, 2005
Advance Access publication January 10, 2006
Associate Editor: Keith A Crandall

ABSTRACT

Summary: BioShell is a suite of programs performing common tasks accompanying protein structure modeling. BioShell design is based on UNIX shell flexibility and should be used as its extension. Using BioShell various molecular modeling procedures can be integrated in a single pipeline.

Availability: BioShell package can be downloaded from its website http://biocomp.chem.uw.edu.pl/BioShell and these pages provide many examples and a detailed documentation for the newest version.

Contact: dgront@chem.uw.edu.pl

INTRODUCTION

Current protein structure prediction and modeling methods are composed of several more or less independent programs. Some protocols for protein modeling have been already fully automated (Sanchez and Sali, 1998; Schwede *et al*., 2003) They are operated by WWW pages and can build a protein model without any human intervention.

In many cases, modules used at different stages of protein modeling are written by different authors. Therefore, one needs an environment to bring them together into a single pipeline. In most cases a scripting language is employed. Other authors have built their own bioinformatics extensions to Perl (Stajich *et al*., 2002) and Python (Hamelryck and Manderick, 2003) languages. MMTSB (Feig *et al*., 2004) tool set contains Perl scripts and programming libraries that wrap-up several standard molecular modeling packages as CHARMM (Brooks *et al*., 1983) and AMBER (Pearlman *et al*., 1995). These full-atom models are combined with a low-resolution simulation program (Skolnick *et al*., 1997) providing a pipeline for multiscale modeling.

The approaches discussed above imply that a user knows how to write Perl or Python scripts. Although such knowledge is very useful, it is not common. A simpler choice is to employ a shell environment as an external interface layer.

High-throughput data processing in an automated fashion is especially important when many proteins are modeled in a similar way. We faced this problem during the last edition of the CASP experiment. BioShell comes from our experiences gained during the long process of multistage modeling of all CASP targets.

A suite of programs presented here that convert file formats, perform simple calculations and data pre- and post-processing in protein structure modeling. We follow the general idea of UNIX-like operating systems: one task–one program. BioShell tools can be easily combined with a scripting language (e.g. Python or Perl) or invoked by shell scripts. In some sense BioShell programs may be treated as new shell 'commands'. Similar approach was applied in Tinker (Pappu *et al*., 1998), which is a tool for molecular modeling. Another example is Emboss (Rice *et al*., 2000) oriented mainly on sequences rather than protein structures.

*To whom correspondence should be addressed.

DETAILS OF BIOSHELL SUITE

Generally, BioShell is a way of performing and integrating computations employing UNIX shell as an interface platform. The package itself is composed of several programs, which extend functionality of shell or scripting languages, as Python or Perl. Most of the examples given in the BioShell website are related to the CABS (Kolinski, 2004) modeling tool. Nevertheless, owing to standard file formats and algorithms, BioShell programs can be applied to virtually any modeling protocol.

seq Reads and writes protein sequences in several formats.

str Reads and writes protein structures in PDB and XYZ formats. It also writes selected fragments of a molecule (subset of residues, chains, etc.).

tra Builds a single trajectory from multiple PDB files and saves it in the TRA (Kolinski, 2004) format. The program can also split a TRA into multiple PDB files. It performs other utility actions and calculations on TRA files.

rms Calculates crmsd or drmsd distance between protein structures. It also writes a PDB file with the target structure (or structures) superimposed with a template.

str_calc Calculates structural properties for a given PDB files as short-range distances (r13,r14,r15) and distance maps.

hist Reads a file with one-dimensional or two-dimensional data and makes a histogram.

palign Computes an alignment for a pair of sequence profiles. Secondary structure information could be taken into account.

clust Performs a hierarchical clustering.

dgs (distance geometry sampler) Calculates a structure from a set of distance constraints. Two optimization protocols could be used: steepest descent minimization or Parallel Tempering Monte Carlo.

EXAMPLE APPLICATION OF BIOSHELL

A brief summary of an example application is presented below, while several examples and an extended documentation are available from the BioShell website.

Recently we published a method for hierarchical clustering of protein models, HCPM (Gront and Kolinski, 2005). The program reads a bunch of PDB files and groups similar structures. The

applicability of HCPM is of course limited by many factors. First, only the two most popular distance measures (crmsd and drmsd) are implemented.

BioShell contains 'clust' program which requires pairwise distances as an imput. The 'rms' computes distances between all PDB files in a current directory. This way these two programs provide the same functionality as HCPM. In BioShell however, one can use his own definitions of similarity functions, allowing for instance clustering models of different chain lengths (e.g. from threading servers). In general, the clustering procedure is not restricted to protein models. One can cluster any data provided a distance function is defined. Computing distances is the slowest step of a clustering procedure and can be easily parallelized.

Below we present a simple BioShell script that performs a clustering of a trajectory from the CABS run. In the first line **tra** tool rescales all structures from internal CABS units to Angstroms and computes all crmsd distances. After clustering (second line), resulting list of clusters is sorted according to the fifth column (i.e. cluster size) and a line corresponding to the best cluster is selected. Starting from the 32nd column, the line contains a list of structures that belong to a given cluster. Additionally the list is stored in a 'frames_id' file. Fourth line converts the TRA format into a PDB formatted trajectory. Fifth line copies all structures, listed by 'frames_id' into a subdirectory.

```
./tra -t=trajectory_file -scale=0.61 -drop_dummies -all_crmsd >crmsd_data
./clust -i=crmsd_data -auto_cutoff -out_tree=tree_file >results
sort -n -k 5 results | tail -1 | cut -c 32- | tr ' ' '\n' >frames_id
./tra -t=trajectory_file -f=sequence.fasta -scale=0.61 -drop_dummies -save_pdb
for i in 'cat frames_id'; do cp $i.pdb ./best_cluster_frames/; done
```

ACKNOWLEDGEMENTS

This work was supported by Polish Ministry of Scientific Research and Information Technology (3 T09A 087 028 and PZB-KBN-088/P04/2003).

Conflict of Interest: none declared.

REFERENCES

Brooks,B.R. *et al*. (1983) A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, **4**, 187–217.

Feig,M. *et al*. (2004) MMTSB Tool Set: enhanced sampling and multiscale modeling methods for applications in structural biology. *J. Mol. Graph. Model.*, **22**, 377–395.

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

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

Kolinski,A. (2004) Protein modeling and structure prediction with a reduced representation. *Acta. Biochim. Pol.*, **51**, 349–371.

Pappu,R.V. *et al*. (1998) Analysis and application of potential energy smoothing for global optimization. *J. Phys. Chem. B*, **102**, 9725–9742.

Pearlman,D.A. *et al*. (1995) AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comput. Phys. Commun.*, **91**, 1–41.

Rice,P. *et al*. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.*, **16**, 276–277.

Sanchez,R. and Sali,A. (1998) Large-scale protein structure modeling of the *Saccharomyces cerevisiae* genome. *Proc. Natl Acad. Sci. USA*, **95**, 13597–13602.

Schwede,T. *et al*. (2003) SWISS-MODEL: an automated protein homology-modeling server. *Nucleic Acids Res.*, **31**, 3381–3385.

Skolnick,J. *et al*. (1997) MONSSTER: a method for folding globular proteins with a small number of distance restraints. *J. Mol. Biol.*, **265**, 217–241.

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