BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 12 2005, pages 2914–2916
doi:10.1093/bioinformatics/bti438

Structural bioinformatics

MolIDE: a homology modeling framework you can click with

Adrian A. Canutescu and Roland L. Dunbrack, Jr.*

Institute for Cancer Research, Fox Chase Cancer Center, 333 Cottman Avenue, Philadelphia, PA 19111, USA

Received on February 23, 2005; revised on April 2, 2005; accepted on April 6, 2005
Advance Access publication April 21, 2005

ABSTRACT

Summary: Molecular Integrated Development Environment (MolIDE) is an integrated application designed to provide homology modeling tools and protocols under a uniform, user-friendly graphical interface. Its main purpose is to combine the most frequent modeling steps in a semi-automatic, interactive way, guiding the user from the target protein sequence to the final three-dimensional protein structure. The typical basic homology modeling process is composed of building sequence profiles of the target sequence family, secondary structure prediction, sequence alignment with PDB structures, assisted alignment editing, side-chain prediction and loop building. All of these steps are available through a graphical user interface. MolIDE's user-friendly and streamlined interactive modeling protocol allows the user to focus on the important modeling questions, hiding from the user the raw data generation and conversion steps. MolIDE was designed from the ground up as an open-source, cross-platform, extensible framework. This allows developers to integrate additional third-party programs to MolIDE.

Availability: http://dunbrack.fccc.edu/molide/molide.php

Contact: rl_dunbrack@fccc.edu

INTRODUCTION

In recent years, molecular modeling has entered mainstream biomedical research, contributing significantly to the understanding of protein function through structure. Currently, ∼50% of proteins of interest can be modeled using a publicly available protein structure as a template. The number of structures increases by several tens of proteins each week and given the strong support for structural genomics projects, the number of available structures will increase rapidly. This will markedly improve the chances of finding a template that has at least 30% identity, which is considered a reasonable threshold for obtaining a quality model (Sanchez *et al*., 2000). Consequently the demand for molecular modeling will also increase. There are databases of models available, such as MODBASE (Pieper *et al*., 2002), as well as modeling servers, but because of uncertainties in automatic alignments, modeling is often performed as an interactive process. This allows the interested user to edit the alignment manually and to pay extra attention to important parts of the predicted structure, such as loops near active sites.

Currently there are many free academic software packages available that address each stage involved in the modeling process. Compiling, setting up input files and writing scripts to allow for data exchange between different programs can be very time consuming and cumbersome. We have created an interactive graphical program, Molecular Integrated Development Environment (MolIDE), that addresses these problems. MolIDE establishes a system in which all these details are handled behind the scenes, allowing the user to focus in an effective manner on the practical modeling tasks that have to be performed. MolIDE integrates sequence database search, sequence-structure alignment, template choice, alignment editing based on structure, secondary structure prediction, loop modeling and side-chain modeling.

*To whom correspondence should be addressed.

METHODOLOGY

MolIDE's interface was organized to follow the logical order of the operations involved in the basic homology modeling procedure. For convenience, during installation each program is configured with a set of default settings and command line parameters. For fine-tuning to a specific application, the most important parameters are accessible through the graphical interface.

A typical session of MolIDE is shown in Figure 1. The user starts by opening a FASTA formatted sequence file (Fig. 1, upper right). Sequence search is performed using our in-house modified version of PSI-BLAST (Altschul *et al*., 1997). This version of PSI-BLAST has been modified (G. Wang and R. Dunbrack, unpublished) to output the profile matrix after each round of searching the non-redundant sequence database (Wheeler *et al*., 2004) to individually named files. That is, a search of N rounds produces a set of $N - 1$ files, named *seqname_chk1*, *seqname_chk2*, . . ., *seqname_chkN-1*. (PSI-BLAST does not output a profile from the last round of the search.) The sequence profiles output after each round of PSI-BLAST are then used to search against the protein sequences deposited in the PDB; the result is a list of hits from the PDB corresponding to each round of non-redundant database search. This allows the user to observe changes in the list of hits or their alignments as more distantly related proteins are added to the profile. The profiles are also used to produce secondary structure predictions from PSIPRED (Jones, 1999). The user can visualize the secondary structure predictions (one for each round of PSI-BLAST) in a color-coded stacked representation (red = helix, green = sheet) in which the intensity of the color reflects the confidence level in the prediction, as provided by PSIPRED (Fig. 1, lower right). This approach permits the user to identify a possible profile drift caused by the inclusion into the profile of unrelated or misaligned sequences. The PDB search results window (Fig. 1, middle right) lists data for each hit in separate columns: the template PDB entry code, the experimental method (X-ray or NMR), resolution, E-value, identities, positives, gaps, start and end residues in both the query sequence and the hit PDB sequence, and alignment length. The user can sort the table by each of

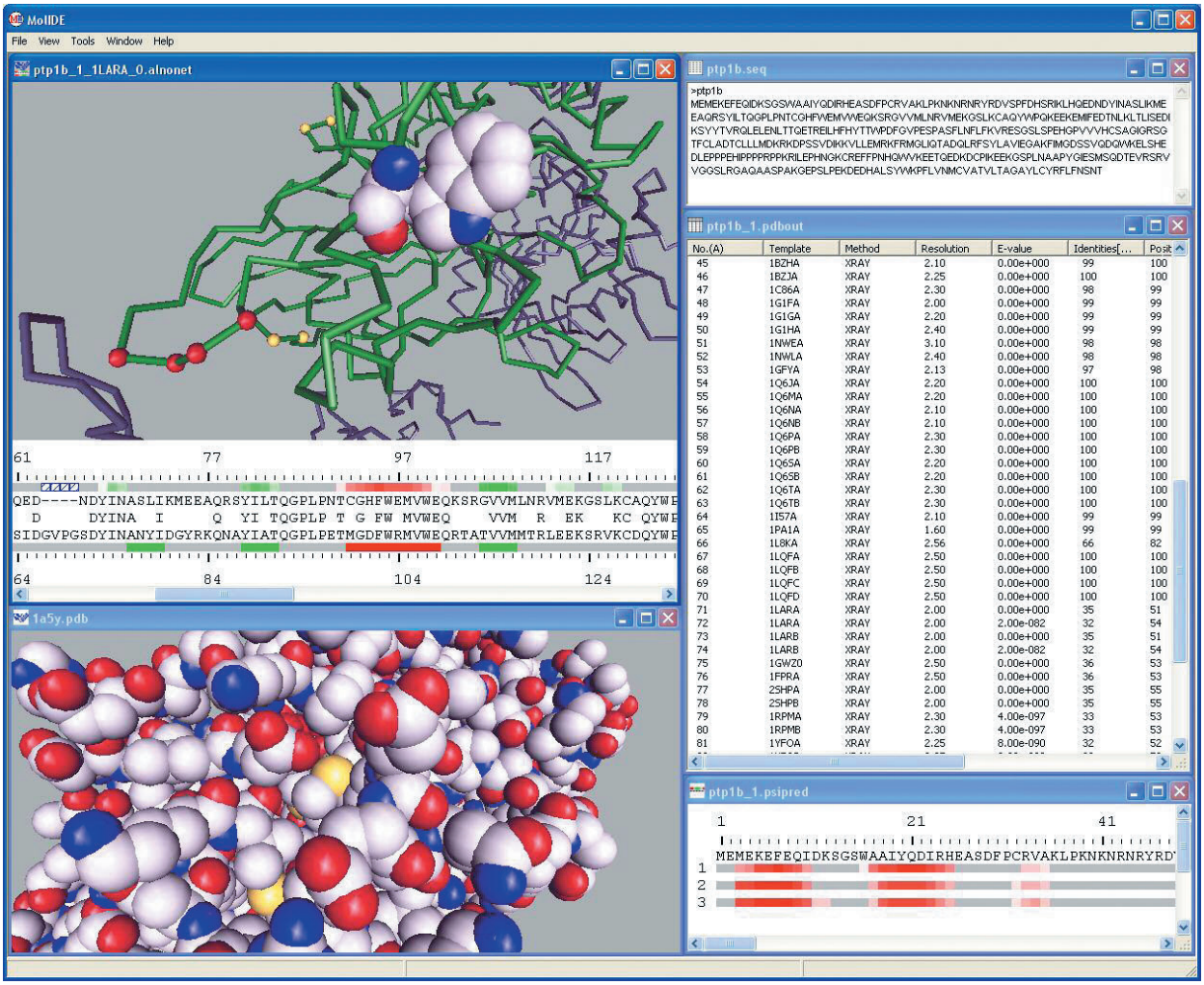

Fig. 1. MolIDE screenshot showing various data representations. The FASTA-formatted target sequence (upper right), the list of hits from a search of the PDB (middle right) and predicted secondary structure (lower right) are shown. In the upper left, the modeling window is shown, including the alignment and associated predicted and experimental secondary structures and a backbone representation of the template structure. In the lower right, a space-filling representation of a protein structure is shown.

these parameters in ascending or descending order just by clicking the column header.

Double-clicking on a certain template in this table automatically obtains the files necessary to build the model based on that template, and opens a new window for editing the alignment (Fig. 1, upper left panel). MolIDE creates a view that contains the sequence alignment with the chosen template, query secondary structure prediction, experimental template secondary structure and a three-dimensional representation of the template protein. A problematic step in modeling by homology is obtaining a correspondence between the sequence numbering used by PSI-BLAST or other alignment programs in which the sequence of the structure runs from 1 to L, the length of the protein, while the numbering of residues in the coordinates may not begin on 1 and/or may skip numbers due to missing electron density or add residues such as “62A”. We obtain the correspondence information from the XML-formatted files from the PDB (http://www.rcsb.org) and compile a file for each PDB entry stored in a database called S2C (J. Arthur, G. Wang, and R.L. Dunbrack, unpublished) that can be downloaded from our website (http://dunbrack.fccc.edu/s2c) and stored locally. Alternatively, if MolIDE cannot find the needed PDB and S2C files in the specified local directories, it will download the template XML-formatted structure file from the RCSB website and extract the PDB and S2C information using XML2PDB, a utility we have created for this purpose (also available separately at http://dunbrack.fccc.edu/xml2pdb.php).

MolIDE provides for easy, visually assisted alignment editing. In this view (Fig. 1, upper left), the protein structure is represented as the backbone and insertions and deletions in the alignment are marked on the structure with red balls on each deleted residue and yellow balls marking the positions flanking the sites of insertions. Gaps can be inserted or deleted from the sequence alignment and the positions of the index markers move along the structure in real time. The predicted and experimental secondary structures, the residue numbers, and the

numbers of identities, similarities, and gaps in the alignment view are instantly updated as gaps are inserted or deleted. In addition, the template structure can be viewed as a space-fill rendering (Fig. 1, lower left).

Once the alignment is edited, the model can be generated by calling loop and side-chain modeling methods from the menus. An initial model is produced from the alignment by copying the backbone coordinates of aligned residues from the template structure, and renumbering and renaming the residues. Side-chain coordinates for conserved residues are also copied to the model. The user can then build side chains for non-conserved residues onto the alignment using the SCWRL program (Canutescu *et al*., 2003) with an option to preserve the Cartesian coordinates of the conserved side chain. Loop building proceeds first by choosing the *N*- and *C*-terminal anchor positions flanking the loop and then calling the program Loopy (Xiang *et al*., 2002). Ordinarily, the anchor positions can be placed on the ends of the whole loop in which an insertion or deletion occurs. If a loop is particularly short, the anchors may need to be placed further apart. If a loop is very long, it may be preferable to model only a few residues on either side of the insertion/deletion point.

MolIDE was designed from the very beginning as a Linux and Windows cross-platform application. Given the requirement of handling multiple data types and views at the same time, the program was written in object-oriented C++, following a multiple document interface (MDI) architecture. The document/view architecture was implemented using the cross-platform wxWindows library (http://www.wxwindows.org). The three-dimensional representation uses the OpenGL library (http://opengl.org) and takes advantage of the optimizations that are available in modern video cards. Significant effort was invested into porting some of the third party computer programs (PSIPRED, LOOPY) from Linux to Windows. Additionally, in order to make our program as user-friendly as possible, we designed an automated setup procedure that installs and generates the configuration files required by each individual program included in the distribution.

The MolIDE source code structure allows for easy additions or modifications by interested users. Within the multiple document interface paradigm, for each action that involves data processing and visualization, there are two classes: a *document class* that handles the data input and processing, and a *view class* that takes care of the representation and the user interaction. For instance, the code that loads and displays a PDB structure is composed of the *CPDBDoc* class (implemented in PDBDoc.cpp file) and the *CPDBView* class (implemented in PDBView.cpp file). This naming convention is common for most of the classes developed for the MolIDE project. Thus, in addition to the very modular and extensible organization of the code, the programmer can identify very easily by the name of a source file the class defined inside, as well as the functional role of that class. The programmer can also very easily use one of the defined classes to derive a class having more functionality. It is even easier to replace functions or add new ones, when there is no visualization associated with that action. For instance, the code responsible for running SCWRL3 is located in three files: OptScwrlDlg.cpp which implements the dialog handling the options for that specific program, ScwrlSettings.cpp which reads and writes from/to the hard disk the options chosen by the user, and RunScwrlDlg.cpp which runs SCWRL3 and displays the program output.

We are confident that MolIDE will provide a convenient interface for basic comparative modeling for both novice and expert alike. We plan to add a number of features that will improve the quality of homology modeling with MolIDE. These include multiple structure alignment of templates, multiple sequence alignments of query homologs from both the non-redundant sequence database and the PDB, model evaluation and modeling from multiple templates.

ACKNOWLEDGEMENTS

We would like to thank the developers of the third party programs that have been incorporated into MolIDE for both their software and for help in porting their programs to windows: Zhexin Xiang, Barry Honig, Liam McGuffin, David Jones, and Stephen Altschul, as well as the NIH and the Pew Charitable Trusts for financial support.

REFERENCES

Altschul,S.F. *et al*. (1997) Gapped blast and psi-blast: a new generation of database programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Canutescu,A.A. *et al*. (2003) A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.*, **12**, 2001–2014.

Jones,D.T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.

Pieper,U. *et al*. (2002) Modbase, a database of annotated comparative protein structure models. *Nucleic Acids Res.*, **30**, 255–259.

Sanchez,R. *et al*. (2000) Protein structure modeling for structural genomics. *Nat. Struct. Biol.*, **7** (Suppl), 986–990.

Wheeler,D.L. *et al*. (2004) Database resources of the national center for biotechnology information: Update. *Nucleic Acids Res.*, **32** (Database issue), D35–D40.

Xiang,Z. *et al*. (2002) Evaluating conformational free energies: the colony enegy and its application to the problem of protein loop prediction. *Proc. Natl Acad. Sci. USA*, **99**, 7432–7437.