

# SWIFT MODELLER v2.0: a platform-independent GUI for homology modeling

Abhinav Mathur · Shankaracharya ·  
Ambarish S. Vidyarthi

Received: 20 September 2011 / Accepted: 21 November 2011 / Published online: 9 December 2011  
© Springer-Verlag 2011

**Abstract** SWIFT MODELLER v2.0 is a platform-independent Java-based graphical user interface to MODELLER. It provides an interactive homology modeling solution by automating the formatting, scripting, and data extraction processes, meaning that the user only needs to paste in the protein target sequence as input. SWIFT MODELLER v2.0 takes a step-by-step approach where the flow of the software screens depicts steps in the homology modeling protocol. Ramachandran plots and DOPE profile graphs are sketched and displayed for in-depth model analysis, along with an embedded Jmol viewer for 3D visualization of the constructed model. SWIFT MODELLER v2.0 is functional on all Linux-based and Microsoft Windows operating systems for which MODELLER has been developed. The software is available as freeware at <http://www.bitmesra.ac.in/swift-modeller/swift.htm>.

**Keywords** MODELLER GUI · MODELLER · Homology modeling · Loop modeling

## Introduction

MODELLER is a computer program for comparative protein structure modeling developed by Andrej Sali [1, 2] at the University of California, San Francisco. MODELLER is controlled via command line arguments, and lacks a

graphical user interface (GUI). It is a complex software protocol that requires extensive study of manuals and tutorials to use it effectively.

Attempts have already been made to ease the use of MODELLER by providing or developing a GUI base, including MINT, EasyModeller [3] and SWIFT MODELLER [4]. MINT was developed for MODELLER 8 and is now a dormant product. Also, it functions exclusively on Linux. EasyModeller and SWIFT MODELLER, on the other hand, require a Microsoft Windows platform. Aside from these, Accelrys makes commercially available GUI for MODELLER.

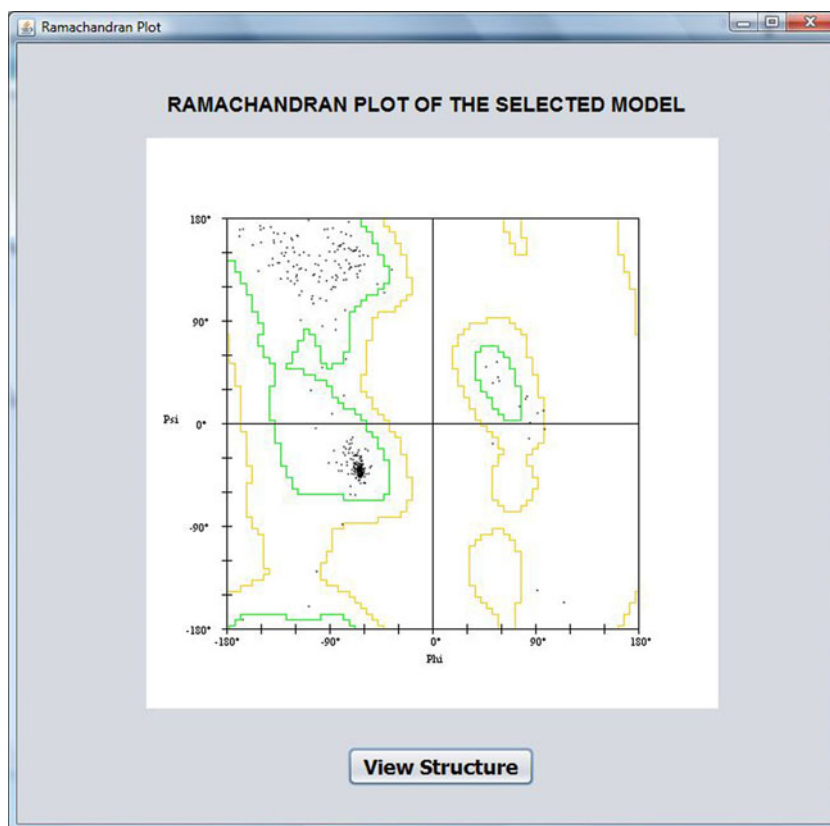
SWIFT MODELLER v2.0, a Java-based GUI to MODELLER, has been developed with a view to providing easy homology modeling to both Windows as well as Linux users, with the same architecture functioning on all systems upon which MODELLER can run.

It automates tiresome steps such as the formatting of input, template protein database searching, writing and executing Python scripts, and meaningful data extraction from output files, thus saving considerable time and human effort when performing homology modeling with MODELLER. In addition, it means that the user only needs to paste in the protein sequence as input, thereby lowering the technical skill level required to use MODELLER [4].

Ramachandran plotting has been added as a new enhancement that, along with the DOPE score profile plotting functionality that was already incorporated into SWIFT MODELLER, can aid the user in comprehensively analyzing the constructed protein model. Also, Jmol, an open-source Java viewer for viewing chemical structures in 3D (<http://www.jmol.org/> [5])—which was treated as external software in the previous version—has been embedded into this enhanced version.

A. Mathur · Shankaracharya · A. S. Vidyarthi (✉)  
Department of Biotechnology,  
Birla Institute of Technology, Mesra,  
Ranchi 835215 Jharkhand, India  
e-mail: [asvidyarthi@bitmesra.ac.in](mailto:asvidyarthi@bitmesra.ac.in)  
URL: <http://www.bitmesra.ac.in/swift-modeller/swift.htm>

**Fig. 1** Screenshot of a Ramachandran plot of a constructed model



## Implementation

### Model construction

SWIFT MODELLER v2.0 takes the target protein sequence and the template(s) as input from the user, edits the required Python files, and executes them via MODELLER using a batch file (Windows platform) or a shell script (Linux platform) at the backend. The user is presented with the option to automatically search for templates conforming to the target sequence in a database bundled along with the software. The selected template(s) are then downloaded by the software itself from the PDB website. The user can also provide specific templates instead of searching for them in the database. Useful data from the output log files are extracted and displayed in the form of meaningful tables and graphs. All files are stored in one folder named after the date and time of the work instance.

### Model analysis

Along with the DOPE score evaluation graphs of the templates and the constructed models (carried forward from the previous version of SWIFT MODELLER), the

current version additionally sketches a Ramachandran plot for the constructed model to aid the user in model analysis. Psi–phi angles for the model in question are calculated using the BIOJAVA API [6]. A Java program for drawing Ramachandran plots, developed by Peter N. Robinson at the Institute of Medical Genetics, Charité University Hospital, Berlin, Germany, has been integrated with the SWIFT MODELLER v2.0 architecture, which plots the psi–phi angle values in a publication-quality image file that is displayed promptly (see Fig. 1).

### 3D visualization of models

The user, after analyzing the model via the energy profile graph and the Ramachandran plot, can visualize the model in a 3D environment using Jmol viewer, which has been embedded into the SWIFT MODELLER v2.0 framework. The model is presented using the “cartoon” scheme by default, and is colored according to the various amino acids that the model consists of. The user can modify the visualization parameters by accessing the Jmol menu by right-clicking in the Jmol window space.

## Changes from SWIFT MODELLER

### Platform independency

SWIFT MODELLER v2.0 is Java-based and thus runs on all GUI-based Windows and Linux-based operating systems for which MODELLER architecture exists. This leads to a large user pool, and provides easy homology modeling.

### Ramachandran plot

The addition of Ramachandran plotting functionality serves as a supplementary aid, along with the pseudo-energy scores, when analyzing the model constructed by MODELLER, and can thus be used to select or discard a model.

### Embedded Jmol

Jmol has been embedded into SWIFT MODELLER v2.0, unlike in the previous version, where it ran as an external application. This has been done to reduce the running time of the GUI flow, and it also reduces the software package size.

### Error handling

Efforts have been made to provide the user with logical explanations for input errors and failed jobs, unlike the previous version, which simply crashed. An error log file is generated, and the software shuts down gracefully.

## Availability and requirements

*Project name* SWIFT MODELLER v2.0

*Project homepage* <http://www.bitmesra.ac.in/swift-modeller/swift.htm>

*Operating systems* See the MODELLER release notes for more information on the tested platforms: <http://www.salilab.org/modeller/release.html>)

*Other requirements* The system must have MODELLER and Python (a version compatible with the MODELLER version) preinstalled, along with Numpy and Matplotlib. Microsoft Windows Vista and 7 users must install MODELLER in a partitioned drive other than C:\ to avoid administrator rights related hassles.

*License* Free to use.

*Any restriction on use by non-academics?* None.

**Acknowledgments** The authors are thankful to the Sub-Distributed Information Center (BTISnet SubDIC), Department of Biotechnology (no. BT/BI/04/065/04), New Delhi, India, for financial support; and the Department of Biotechnology, Birla Institute of Technology, Mesra, Ranchi, for providing us with the infrastructure and facilities for the present study.

## References

1. Šali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol* 234:779–815
2. Fiser A, Do RK, Šali A (2000) Modeling of loops in protein structures. *Protein Sci* 9:1753–1773
3. Kuntal BK, Aparoy P, Reddanna P (2010) EasyModeller: a graphical interface to MODELLER. *BMC Res Notes* 3:226–330
4. Mathur A, Shankaracharya VAS (2011) SWIFT MODELLER: a JAVA based GUI for molecular modeling. *J Mol Model*. doi:10.1007/s00894-011-0960-4
5. Herráez A (2006) Biomolecules in the computer: Jmol to the rescue. *Biochem Mol Biol Edu* 34:255–261
6. Holland RCG, Down T, Pocock M, Prlić A, Huen D, James K, Foisy S, Dräger A, Yates A, Heuer M, Schreiber MJ (2008) BioJava: an Open-Source Framework for Bioinformatics. *Bioinformatics* 24:2096–2097