

PyMod 2

User's Guide

PyMod 2 Documentation
(Last updated: 7/11/2016)

<http://schubert.bio.uniroma1.it/pymod/index.html>

Department of Biochemical Sciences “A. Rossi Fanelli”, Sapienza University of Rome, Italy

2016: Giacomo Janson, Chengxin Zhang, Alessandro Paiardini

2011-2015: Emanuele Bramucci, Alessandro Paiardini, Francesco Bossa, Stefano Pascarella

For more information, please contact: giacomo.janson@uniroma1.it

Table of Contents

1 Introduction	1
1.1 PyMod 2 Overview	1
1.2 About this User Manual	1
2 Installation Instructions	3
2.1 Windows	4
2.2 Mac OS X	5
2.3 Linux	6
2.3.1 Install NumPy and Biopython, ClustalW/Omega, MUSCLE and the BLAST+ suite	6
2.3.2 Install PyMod 2 and other missing components using the PyMod Installer Bundle	6
2.3.3 Install MODELLER	7
3 PyMod 2 Tools	8
3.1 BLAST	8
3.2 PSI-BLAST	8
3.2.1 Configuration within PyMod	8
3.3 ClustalW	9
3.4 Clustal Omega	9
3.5 MUSCLE	10
3.6 ESPript 3	10
3.7 WebLogo 3	10
3.8 CAMPO	11
3.9 PSIPRED	11
3.9.1 Installation	11
3.9.2 Configuration within PyMod	12
3.9.3 Note on sequence databases to be used with PSIPRED	12
3.10 Ksdssp	12
3.11 MODELLER	13
4 PyMod 2 Functionalities	14
4.1 Run PyMod 2	14
4.2 First-time Usage	14
4.3 Begin a PyMod Project	14
4.4 Save and Load PyMod Projects	15
4.5 Input and Output	15
4.5.1 Input sequence files	15
4.5.2 Input structure files	16
4.5.3 Input alignment files	16
4.5.4 Output sequence files	17
4.6 Analyzing Sequences and Structures	17
4.6.1 PyMod main menu	17
4.6.2 Sequences	18
4.6.3 Structures	18
4.6.4 Alignments and clusters	19

4.7 Similarity Searches	19
4.7.1 Using NCBI-BLAST	19
4.7.2 Using PSI-BLAST	20
4.7.3 Import hit sequences in PyMod	22
4.7.4 Fetching structures over the PDB	22
4.7.5 Expanding clusters	23
4.8 Alignment Building	24
4.8.1 Sequence alignments	24
4.8.2 Structural alignments	24
4.8.3 Profile alignments	25
4.8.4 Additional alignment modes	25
4.9 Alignment Analysis	26
4.9.1 Sequence clusters and alignments	26
4.9.2 Matrices menu	27
4.9.3 Trees menu	28
4.9.4 Evolutionary conservation menu	29
4.9.5 Alignment rendering menu	29
4.10 Sequence and Structure Analysis	30
4.10.1 PSIPRED	30
4.10.2 Ramachandran plot	31
4.10.3 Assess DOPE	32
4.11 Homology Modeling Using MODELLER	32
4.11.1 Building homology models	33
4.11.2 Single chain homology modeling	33
4.11.3 Multiple chain homology modeling	33
4.11.4 The Homology Modeling Window	34
4.11.5 Main tab	34
4.11.6 Template complex selection in the Main tab (multiple chain homology modeling mode only)	34
4.11.7 Modeling options in the Main tab	35
4.11.8 Template selection in the Main tab	35
4.11.9 Disulfides tab	36
4.11.10 Options tab	37
4.11.11 Analyzing results	40
4.12 PyMod 2 Options	42
4.13 BLAST Sequence Databases	42
4.13.1 Add a BLAST Sequence Database to PyMod 2	42
5 Usage Examples	44
5.1 Building a Single Chain Homology Model: Modeling of the Dihydrofolate Reductase from Mycobacterium avium	44
5.1.1 Launch PyMod 2 and import the target sequence	44
5.1.2 Template searching and selection	45
5.1.3 Alignment building and editing	49
5.1.4 Model building	50
5.1.5 Model analysis	53

5.2 Building a Multiple Chain Homology Model: Modeling of Human Hemoglobin Portland I	56
5.2.1 Launch PyMod and import the target sequences	56
5.2.2 Import the template complex in PyMod	58
5.2.3 Target-template alignments building	59
5.2.4 Model building	61
5.2.5 Model analysis	62
6 PyMod 2 Development	65
6.1 General Information and Contacts	65
6.2 List of Tested Platforms	65
7 Appendix	66
7.1 Locate PyMOL Default Plugin Directory	66
7.2 Manually Install PyMod 2	66
7.2.1 For PyMOL versions with a Plugin Manager (PyMOL >= 1.5.0.5)	66
7.3 PyMod Installer Bundle	67
7.3.1 PyMod Installer Bundle Usage	67
7.3.2 install_all.py	68
7.3.3 get_information.py	70
8 Bibliography	71

1 Introduction

1.1 PyMod 2 Overview

PyMod 2 [Janson et al., 2016] is a PyMOL [Schrödinger, 2015] plugin, designed to act as an intuitive interface between PyMOL and several bioinformatics tools (see **Figure 1.1**). PyMod original version, PyMod 1.0 [Bramucci et al., 2012], was developed with the aim of creating a simple, yet powerful tool for sequence and structure analysis and homology modeling within PyMOL. The current PyMod release, PyMod 2, has been extended with a rich set of functionalities that substantially improve it, particularly in its ability to build homology models through the popular MODELLER package [Sali and Blundell, 1993]. Starting from the amino acid sequence of a target protein, users may take advantage of PyMod 2 to carry out the three steps of the homology modeling process (that is, template searching, target-template sequence alignment and model building) in order to build a 3D atomic model of a target protein (or protein complex). Additionally, thanks to its flexibility, PyMod 2 may also be used outside the homology modeling context, in order to extend PyMOL with numerous types of functionalities. Sequence similarity searches, multiple sequence-structure alignments and evolutionary conservation analyses can all be performed in the PyMod 2/PyMOL environment (see **Figure 1.1**).

PyMod 2 shows how the possibility to perform different kind of bioinformatic analyses in a single integrated environment can greatly facilitate biological data analysis and the building of hypotheses which can be used to help guide biological research projects.

1.2 About this User Manual

This manual contains in depth information on how to install and use PyMod 2 on your system. If you are using PyMod 2 for the first time you may want to begin by reading:

- **Installation Instructions**: in order to be able to install PyMod 2 on your system.
- **First-time Usage**: in order to configure PyMod 2 before starting to use it.
- **Usage Examples**: in order to learn how to use PyMod 2 by following one among a series of tutorials.

If you want more information about a specific PyMod 2 functionality, you may find it in the **PyMod 2 Functionalities** section of this manual.

If you want to participate in PyMod 2 development you may find useful information in the **PyMod 2 Development** section.

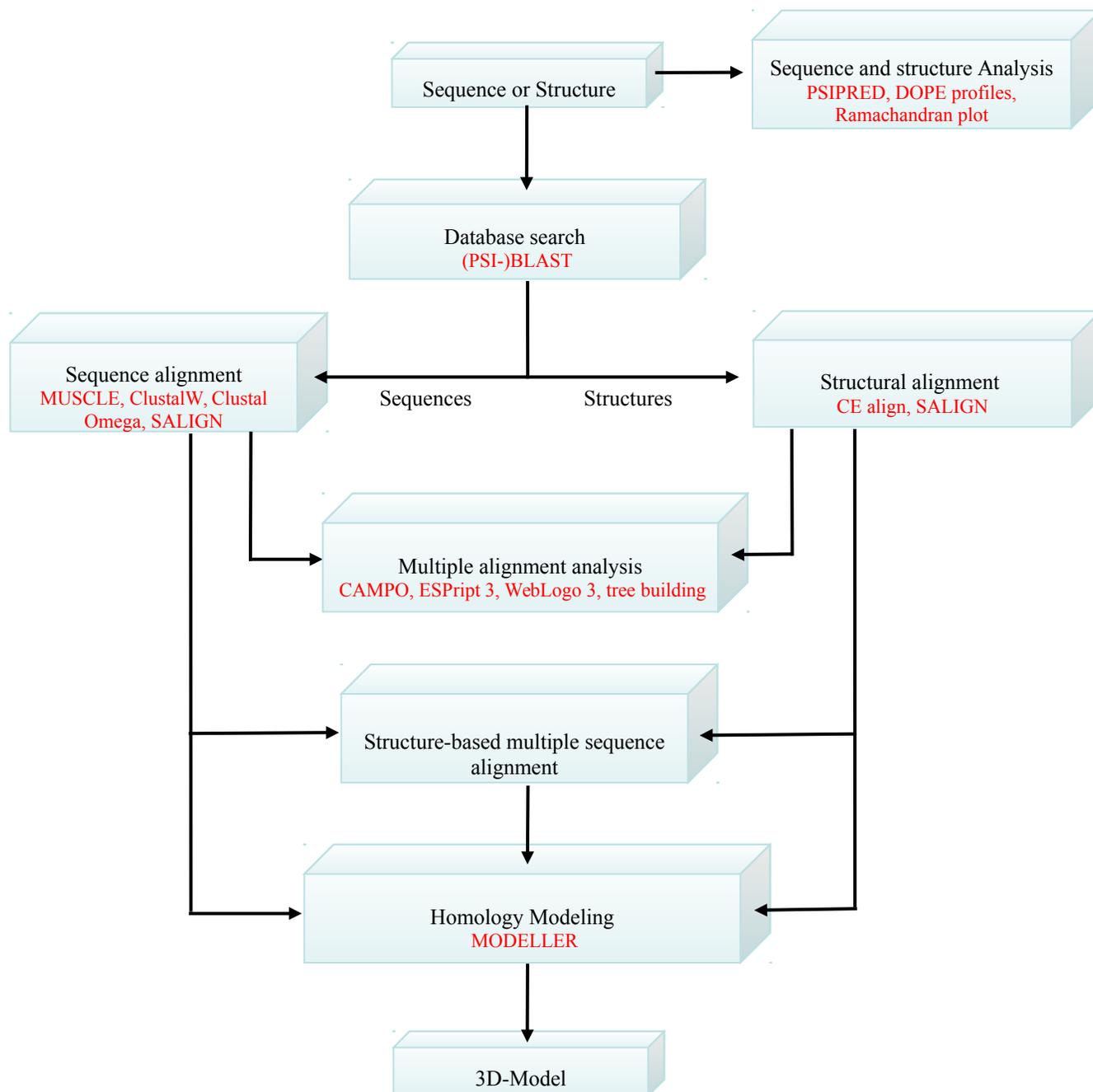


Figure 1.1. Flowchart representing PyMod 2 workflow. Every step can be considered as standalone, e.g. you don't need to use BLAST (for sequence retrieving) before aligning (for example with Clustal Omega) two or more sequences. The algorithms which can be used in PyMod 2 are highlighted in red.

2 Installation Instructions

PyMod 2 runs on Windows, Mac OS X and Linux versions of PyMOL. It is compatible with PyMOL version ≥ 0.99 and it has been tested on both unofficial and official PyMOL builds distributed by Schrödinger (for more information on compatibility, see [List of Tested Platforms](#)). Its only core Python dependencies are NumPy [van der Walt et al., 2011] and Biopython [Cock et al., 2009]. However, PyMod 2 itself without any external third party tools (see [Table 2.1](#) and [PyMod 2 Tools](#)), will not provide full functionalities. In order to use additional functionalities, you will need to install [Table 2.1](#) bioinformatics tools on your system. To facilitate the installation of PyMod 2 and its components, we provide easy-to-use installer packages (see [PyMod Installer Bundle](#)). If you prefer to perform a manual installation, you may install PyMod 2 as any regular PyMOL plugin and use your own versions of [Table 1](#) tools. In the sections below we will cover the installation of PyMod 2 and all its features on today's major operating systems using the [PyMod Installer Bundle](#).

Package	Installation Required to Run PyMod 2	Use within PyMod 2	Installable through the PyMod Installer Bundle	Reference
Python libraries				
NumPy	Required	Core PyMod dependency	Yes	[Van der Walt et al., 2011]
Biopython	Required	Core PyMod dependency	Yes	[Cock et al., 2009]
External tools				
BLAST+ suite	Optional	Implements PSI-BLAST, required by PSIPRED	Yes	[Camacho et al., 2009]
ClustalW	Optional	Build multiple sequence alignments, build distance trees	Yes	[Thompson et al., 1994]
Clustal Omega	Optional	Build multiple sequence alignments	Yes	[Sievers et al., 2011]
MUSCLE	Optional	Build multiple sequence alignments	Yes	[Edgar, 2004]
CE alignment	Built-in support since PyMOL version 1.5	Build multiple structural alignments	Yes, but not needed since PyMOL version 1.5	[Shindyalov and Bourne, 1998]
PSIPRED	Optional	Secondary structure predictions	Yes	[Jones, 1999]
Ksdssp	Optional	Secondary structure assignment	Yes	[Kabsch and Sander, 1983]
MODELLER	Optional	Build homology models, build multiple sequence and structural alignments through the SALIGN module	Yes, on Windows and Mac OS X	[Sali and Blundell, 1993]
In house algorithms				
CAMPO	Built-in	Highlight conserved regions in multiple sequence alignments	Not needed	[Paiardini et al., 2005]
Web Services				
NCBI BLAST	Accessible via internet connection	Implements BLAST	Not needed	[Altschul et al., 1997]
WebLogo 3.0	Accessible via internet connection	Build sequence logos	Not needed	[Crooks et al., 2004]
ESPrpt 3	Accessible via internet connection	Alignment rendering	Not needed	[Robert and Gouet, 2014]

Table 2.1. Python libraries on which PyMod 2 depends and external tools with which PyMod 2 can interface with.

2.1 Windows

- In order to install PyMod 2 and its components on your system, you can use the **install_all.py** script shipped in the **PyMod Installer Bundle**. This script will automatically install PyMod 2, NumPy, Biopython, ClustalW, Clustal Omega, MUSCLE, PSI-BLAST, PSIPRED, ksdssp, MODELLER and the SwissProt and PDB sequence databases on your system and configure them for usage within PyMod 2. If any of the previous components is already installed on your system, the script will not reinstall it, but it will configure PyMod 2 in order to use their already existing version.
- To use the **install_all.py** script, download the latest version of the **PyMod Installer Bundle** zip file at <http://schubert.bio.uniroma1.it/pymod/download.html>. Make sure to choose the **Windows** version.
- Once you have downloaded the zip file, unzip its content in any folder of your system (right click on the file and use the *Extract All...* command). In the unzipped directory you will find some Python scripts (files ending in .py) and a zipped file containing all the installation files.
- To run the **install_all.py** script, launch PyMOL and use the *File* → *Run...* command from PyMOL main menu. Select the **install_all.py** file in the **PyMod Installer Bundle** directory you unzipped before (for details see **Launch Installer Bundle Scripts**). The installation process will then begin.
- If you do not have a MODELLER installation on your system, the script will automatically download in the Installer Bundle directory the Windows MODELLER installer (from https://salilab.org/modeller/download_installation.html). The script will also launch the MODELLER installer (please note that on some Windows version, the MODELLER installer window will not automatically open, but you will have to click on its minimized element in the taskbar of your desktop). In the MODELLER installation process you will have to input your registration key (which you can obtain from: <https://salilab.org/modeller/registration.html>). If you decide to skip MODELLER installation, you may perform it later by yourself at any time.
- When the installation process is completed, the '*Installation successful*' message showed in **Figure 2.1** will appear in the output region.

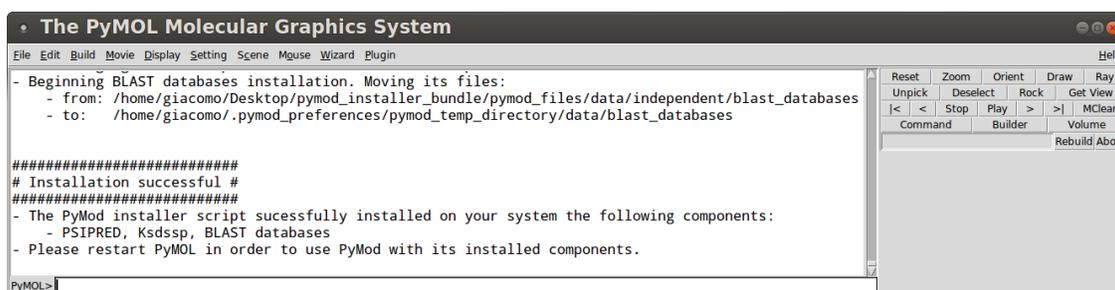


Figure 2.1. When the installation process is completed successfully, the message shown in the figure will appear in PyMOL main window output region.

- In order to use PyMod 2 (and all the newly installed components), restart PyMOL and launch PyMod 2 from PyMOL plugin menu.

2.2 Mac OS X

- In order to install PyMod 2 and its components on your system, you can use the **install_all.py** script shipped in the **PyMod Installer Bundle**. This script will automatically install PyMod 2, NumPy, Biopython, ClustalW, Clustal Omega, MUSCLE, PSI-BLAST, PSIPRED, ksdssp, MODELLER and the SwissProt and PDB sequence databases on your system and configure them for usage within PyMod 2. If any of the previous components is already installed on your system, the script will not reinstall it, but it will configure PyMod 2 in order to use their already existing version.
- To use the **install_all.py** script, download the latest version of the **PyMod Installer Bundle** zip file at <http://schubert.bio.uniroma1.it/pymod/download.html>. Make sure to choose the **Mac OS X** version.
- Once you have downloaded the zip file, unzip its content in any folder of your system. In the unzipped directory you will find some Python scripts (files ending in .py) and a zipped file containing all the installation files.
- To run the **install_all.py** script, launch PyMOL and use the **File** → **Run...** command from PyMOL main menu (note that on some PyMOL versions this command is substituted by an equivalent **File** → **Run Script...**). Select the **install_all.py** file in the **PyMod Installer Bundle** directory you unzipped before (for details see **Launch Installer Bundle Scripts**). The installation process will then begin.
- If you do not have a MODELLER installation on your system, the script will automatically download in the Installer Bundle directory the Mac OS X MODELLER installer (from https://salilab.org/modeller/download_installation.html). The script will also launch the MODELLER installer. In the MODELLER installation process you will have to input your

registration key (which you can obtain from: <https://salilab.org/modeller/registration.html>). If you decide to skip MODELLER installation, you may perform it later by yourself at any time.

- When the installation process is completed, the '*Installation successful*' message showed in **Figure 2.1** will appear in the output region.
- In order to use PyMod 2 (and all the newly installed components), restart PyMOL and launch PyMod 2 from PyMOL plugin menu.

2.3 Linux

2.3.1 Install NumPy and Biopython, ClustalW/Omega, MUSCLE and the BLAST+ suite

- On many Linux distributions, you can install PyMod 2 dependencies and most of its external tools using your package manager. Depending on your distribution, use one the following commands.
 - For Ubuntu 12.04, 14.04 and 16.04 (and similarly for other Debian based distributions):

```
sudo apt-get install python-numpy python-biopython clustalw clustalo
muscle ncbi-blast+
```

- If your Linux distribution lacks some of these packages (or if your PyMOL build is not able to use the system wide Python modules installed through your package manager, such as in the case of some old PyMOL builds from Schrödinger), you may install them using the **install_all.py** script shipped in the **PyMod Installer Bundle** (see chapter below), or alternatively you may install them manually.

2.3.2 Install PyMod 2 and other missing components using the PyMod Installer Bundle

- In order to install PyMod 2 on your PyMOL build you can use the **install_all.py** script shipped in the **PyMod Installer Bundle**. The script will also install and configure for usage within PyMod 2 PSIPRED, kdsdp and the SwissProt and PDB sequence databases (and also any other missing components which you were not able to install with your package manager). If any of the previous components is already installed on your system, the script will not reinstall it, but it will configure PyMod 2 in order to use their already existing version.
- In order to use the **install_all.py** script, download the latest version of the **PyMod Installer Bundle** zip file at <http://schubert.bio.uniroma1.it/pymod/download.html>. Make sure to choose the **Linux** version.

- Once you have downloaded it, unzip its content in any folder of your system. In the unzipped directory you will find some Python scripts (files ending in .py) and a zipped file containing all the installation files.
- To launch the **install_all.py** script, use the *File* → *Run...* command from PyMOL main menu and select its file in the Installer Bundle directory you unzipped (for details see **Launch Installer Bundle Scripts**). The installation process will then begin.
- When the installation process is completed, the '*Installation successful*' message showed in **Figure 2.1** will appear in the output region.
- In order to use PyMod 2 (and all the newly installed components), restart PyMOL and launch PyMod 2 from PyMOL plugin menu.

2.3.3 Install MODELLER

- PyMod 2 provides a GUI for MODELLER. In order to use it within PyMod 2 on Linux, you will have to install it manually. MODELLER installation on Linux is usually straightforward (it involves only one or few command line commands) and easy-to-use .deb or .rpm packages are available.
- Go to https://salilab.org/modeller/download_installation.html, download the right package for your system and use their well documented installation guides to install MODELLER on your computer. Remember to register (<https://salilab.org/modeller/registration.html>) to get a license key in order to use MODELLER.

3 PyMod 2 Tools

PyMod 2 acts as an interface between PyMOL and numerous popular bioinformatics packages. Here you will find a list of the packages currently integrated in PyMod 2/PyMOL and instructions on how to use them within this environment.

3.1 BLAST

The BLAST algorithm [Altschul et al., 1997] is an heuristic program, which means that it relies on some smart shortcuts to perform the search faster. BLAST performs “local” alignments. Most proteins are modular in nature, with functional domains often being repeated within the same protein as well as across different proteins from different species. The BLAST algorithm is tuned to find these domains or shorter stretches of sequence similarity.

Website: <http://blast.ncbi.nlm.nih.gov/Blast.cgi>

Configuration within PyMod: you can perform a BLAST search within PyMod 2 by connecting to the NCBI BLAST server. In order to do that, you only need an internet connection.

3.2 PSI-BLAST

Position-Specific Iterated (PSI)-BLAST [Altschul et al., 1997] is a more sensitive algorithm with respect to the original BLAST, making it useful for finding very distantly related proteins or new members of a protein family. Use PSI-BLAST when your standard protein-protein BLAST search either failed to find significant hits. The first round of PSI-BLAST is a standard protein-protein BLAST search. The program builds a position-specific scoring matrix (PSSM or profile) from a multiple alignment of the sequences returned with Expect values better (lower) than the inclusion threshold (default=0.005). The PSSM will be used to evaluate the alignment in the next iteration of search. Any new database hits below the inclusion threshold are included in the construction of the new PSSM. A PSI-BLAST search is said to have converged when no more matches to new database sequences are found in subsequent iterations.

Website: <http://blast.ncbi.nlm.nih.gov/Blast.cgi>

Download from: <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>

3.2.1 Configuration within PyMod

If you used the **PyMod Installer Bundle** to install PyMod 2, or if a PSI-BLAST executable is already present on your system (that is, if you have already installed by yourself the BLAST+ suite), PyMod 2 will automatically localize the program's executable file in order to use it. If your PSI-BLAST executable is located in any non-standard location on your system, PyMod 2 will not be able to localize it. In this case you can specify the path of its executable file directory by using the the **PyMod 2**

Options window.

To perform a PSI-BLAST search in PyMod 2, you will need to have some sequence databases on your system. If you installed PyMod 2 using the **PyMod Installer Bundle**, the SwissProt and PDB sequence databases will be already present on your system and PyMod 2 will be already configured to use them. If you want to add additional databases or if you did not use the **PyMod Installer Bundle** to install PyMod 2, please refer to **Add a BLAST Sequence Database to PyMod 2**.

3.3 ClustalW

ClustalW [Thompson et al., 1994] is a general purpose multiple sequence alignment program for DNA or proteins. It attempts to calculate the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen.

Website: <http://www.clustal.org/clustal2/>

Configuration within PyMod: if you used the **PyMod Installer Bundle** to install PyMod 2, or if ClustalW was already installed on your system when you installed PyMod 2, the plugin will automatically localize the program's executable file in order to use it. Thus, no configuration will be needed. If you have your own installation of ClustalW, but its executable file is located in any non-standard location on your system, PyMod 2 will not be able to localize it. In this case you can specify the path of ClustalW executable file path by using the the **PyMod 2 Options** window.

3.4 Clustal Omega

Clustal Omega [Sievers et al., 2011] is a general purpose multiple sequence alignment (MSA) program for protein and DNA/RNA. It produces high quality MSAs and is capable of handling data-sets of hundreds of thousands of sequences in reasonable time. Clustal-Omega uses HMMs for the alignment engine, based on the HAlign package [Söding, 2005]. Guide trees are made using an enhanced version of mBed [Blackshields et al., 2010] which can cluster very large numbers of sequences in $O(N \cdot \log(N))$ time. Multiple alignment then proceeds by aligning larger and larger alignments using HAlign, following the clustering given by the guide tree.

Website: <http://www.clustal.org/omega/>

Configuration within PyMod: if you used the **PyMod Installer Bundle** to install PyMod 2, or if Clustal Omega was already installed on your system when you installed PyMod 2, the plugin will automatically localize the program's executable file in order to use it. Thus, no configuration will be needed. If you have your own installation of Clustal Omega, but its executable file is located in any non-standard location on your system, PyMod 2 will not be able to localize it. In this case you can specify the path of Clustal Omega executable file path by using the the **PyMod 2 Options** window.

3.5 MUSCLE

MUSCLE [Edgar, 2004] is a program for creating multiple alignments of amino acid or nucleotide sequences. A range of options is provided that give you the choice of optimizing accuracy, speed, or some compromise between the two.

Website: <http://www.drive5.com/muscle/>

Configuration within PyMod: if you used the **PyMod Installer Bundle** to install PyMod 2, or if MUSCLE was already installed on your system when you installed PyMod 2, the plugin will automatically localize the program's executable file in order to use it. Thus, no configuration will be needed. If you have your own installation of MUSCLE, but its executable file is located in any non-standard location on your system, PyMod 2 will not be able to localize it. In this case you can specify the path of MUSCLE executable file path by using the the **PyMod 2 Options** window.

3.6 ESPrpt 3

ESPrpt (Easy Sequencing in PostScript) [Robert and Gouet, 2014] is a program which renders sequence similarities and secondary structure information from aligned sequences for analysis and publication purpose. Once users have built a multiple sequence or structural alignment in PyMod 2, they can load in their web browser the ESPrpt page with an alignment file containing the sequences aligned in PyMod. In this way they may readily proceed to render their alignment using the ESPrpt web interface. If the alignment cluster contains any sequence with an associated structure loaded in PyMOL, users may choose to upload the structure PDB file to the ESPrpt web server, so that its secondary structure may also be rendered.

Website: <http://esprpt.ibcp.fr/ESPrpt/ESPrpt/>

Configuration within PyMod: you only need an internet connection to access ESPrpt web page from PyMod 2.

3.7 WebLogo 3

WebLogo [Crooks et al., 2004] is a web based application designed to make the generation of sequence logos easy and painless. Logos are graphical representations used to describe both precisely and concisely the variability occurring at each column of a multiple sequence alignment, thus revealing evolutionary conserved patterns. Once users have built a multiple sequence or structural alignment in PyMod 2, they can access the WebLogo 3 web server in order to build a logo of the alignment. The logo image will automatically open in their default document viewer.

Website: <http://weblogo.threeplusone.com/>

Configuration within PyMod: you only need an internet connection to access WebLogo 3 server from PyMod 2.

3.8 CAMPO

CAMPO [Paiardini et al., 2005] is an algorithm tool that enables the assessment of the evolutionary conservation grade of protein residues. Usually, the evolutionary conservation grade is a useful measure of the importance of a residue. The evolutionary conservation grade can be determined by the variability of residues in the columns of a multiple sequence alignment of homologous proteins. CAMPO assigns a score to each column of a multiple sequence alignment throughout the application of a user-defined mutational matrix and incorporates a weight based on the percentage of sequence identity between proteins being compared. The results obtained can be mapped onto a reference protein structure to allow the identification of functionally important residues and surface regions.

Website: <http://schubert.bio.uniroma1.it/>

Configuration within PyMod: CAMPO is built-in in PyMod 2 and does not need any configuration.

3.9 PSIPRED

PSIPRED [Jones, 1999] is a program used predict protein secondary structure. The idea of this method is to use the information of the evolutionarily related proteins to predict the secondary structure of a new amino acid sequence. PSI-BLAST is used to find homologous sequences and to build a position-specific scoring matrix. This matrix is processed by a two feed-forward neural network, which was constructed and trained to predict the secondary structure of the input sequence. The method is well validated. At the time of original publication, a stringent cross-validation procedure showed the method to be capable of achieving an average accuracy of 76.5% and improvements have been made since that benefit from the increase in available data.

Website: <http://bioinf.cs.ucl.ac.uk/psipred/>

Download from: <http://bioinfadmin.cs.ucl.ac.uk/downloads/psipred/>

3.9.1 Installation

There are two ways in which you can install PSIPRED in order to use it within PyMod 2.

3.9.1.A *Install it using the PyMod Installer Bundle*

You can install PSIPRED using by using the **install_all.py** script shipped in the **PyMod Installer Bundle**. This script will also install PSI-BLAST (in case you have not installed it previously), the SwissProt sequence database (which PSIPRED will use by default) and automatically configure PyMod 2 to use PSIPRED.

3.9.1.B *Install it manually*

You can retrieve PSIPRED source code (see the link above) in order to compile and install it on your system (see the README file inside its .tar.gz file).

3.9.2 Configuration within PyMod

Depending on how you installed PSIPRED, you might need to configure PyMod 2 in order to use PSIPRED within the plugin.

3.9.2.A Installation through the PyMod Installer Bundle

In this case PSIPRED will be already configured and ready for use when you launch PyMod 2.

3.9.2.B Manual installation

If you installed PSIPRED manually, you should first configure PyMod 2 in order to use it. PSIPRED depends on PSI-BLAST, so you will need to install and configure it too (see **PSI-BLAST**). To configure PSIPRED, you will need to specify in the **PyMod 2 Options** window:

- the path of the folder in which its executable files exist (called *bin* in the default PSIPRED installation) in the **Executable Directory** entry.
- the path of its data folder (called *data* in the default PSIPRED installation) in the **Data Files Directory**.
- a path of a BLAST sequence database (see how to do it by reading **Add a BLAST Sequence Database to PyMod 2**) in the **PSIPRED → BLAST Db Directory** entry.

3.9.3 Note on sequence databases to be used with PSIPRED

If you installed PSIPRED using the **PyMod Installer Bundle**, the sequence database used by PSI-BLAST when performing a PSIPRED prediction will be by default the SwissProt one. However, for an optimal prediction, PSIPRED authors recommend to use the UNIREF90 (<http://www.uniprot.org/help/uniref>) one. We ship the PyMod Installer Bundle with the SwissProt database because of its smaller size (at the present time it contains 463,486 sequences, its .tar.gz archive is 123 megabytes) compared to the UNIREF90 one (currently it contains 40,253,516 sequence, its .tar.gz archive is 9.2 gigabytes). Including the UNIREF90 database in PyMod Installer Bundle, would make its package file too large. However (even if we did not carry out large scale benchmarks) in our experience using the smaller SwissProt database, results in PSIPRED predictions similar to those obtained using the UNIREF90 database for many protein targets. If you wish to use the UNIREF90 database for PSIPRED predictions in PyMod 2, you may retrieve the database by yourself (we provide it at this address: <http://schubert.bio.uniroma1.it/downloads/databases/uniref90.tar.gz>) and set it as the default one when performing a PSIPRED prediction in PyMod 2 (see the **Manual installation** section above on how to configure a default database for PSIPRED in PyMod).

3.10 Ksdssp

Ksdssp is an open source implementation of the Kabsch and Sander algorithm for defining the secondary structure of proteins, as described in [Kabsch and Sander, 1983]. It provides a more reliable

algorithm to define the secondary structures of protein structures loaded in PyMod 2/PyMOL with respect DSS, PyMOL default algorithm for secondary structure assignment (<http://www.pymolwiki.org/index.php/Dss>).

Website: <https://www.cgl.ucsf.edu/Overview/software.html#ksdssp>

Configuration within PyMod: if you used the **PyMod Installer Bundle** to install PyMod 2, the plugin will automatically localize the ksdssp executable file in order to use it. Thus, no configuration will be needed. If you have your own copy of ksdssp executable and want to use it within PyMod 2, you can specify its path by using the the **PyMod 2 Options** window.

3.11 MODELLER

MODELLER [Sali and Blundell, 1993] is used for homology of protein three-dimensional structures. The user provides an alignment of a sequence to be modeled with known related structures and MODELLER automatically calculates a model containing all non-hydrogen atoms. MODELLER implements comparative protein structure modeling by satisfaction of spatial restraints, and can perform many additional tasks, including *de novo* modeling of loops in protein structures, optimization of various models of protein structure with respect to a flexibly defined objective function, multiple alignment of protein sequences and/or structures through the SALING [Madhusudhan et al., 2009] module (for which PyMod 2 provides a GUI), clustering, searching of sequence databases and comparison of protein structures.

Website: <https://salilab.org/modeller/>

Configuration within PyMod: if you have installed MODELLER on your system, PyMod 2 will be able to automatically recognize it, so that you may use it within the plugin.

4 PyMod 2 Functionalities

4.1 Run PyMod 2

You can launch PyMod 2 just as any other PyMOL plugin. Once you have installed it, use the *Plugin* → *PyMod 2.X* option in PyMOL's main window (see **Figure 4.1**).

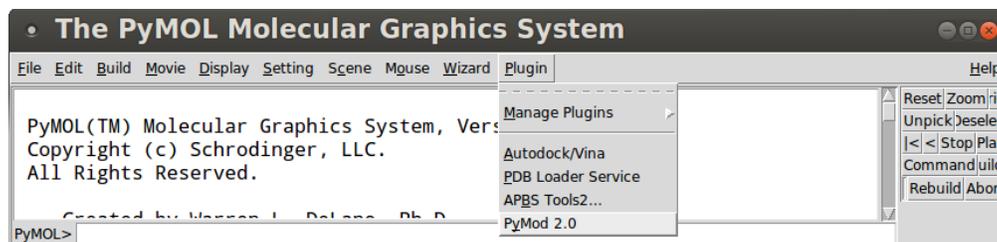


Figure 4.1. You can launch PyMod 2 through the PyMOL *Plugin* submenu.

4.2 First-time Usage

The first time you run PyMod 2 you will be presented with a window (see **Figure 4.2**) asking you to select the folder on of your system in which you would like to create the **PyMod Directory** (by default the target path is set to your home directory). The **PyMod Directory** will contain all the files created during a PyMod 2 session. It will also store the external tools files and data files installed through the **PyMod Installer Bundle**. Once you select it, you may start to use PyMod 2 and start your first project.

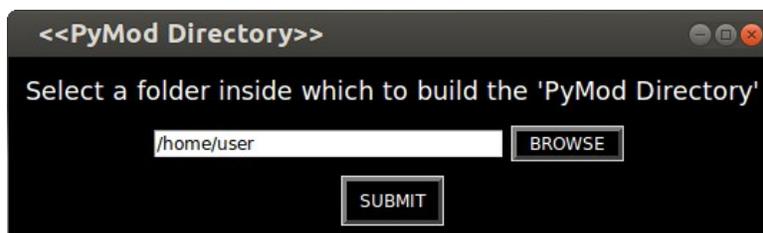


Figure 4.2. Window to select the folder in which the **PyMod Directory** will be created.

4.3 Begin a PyMod Project

Every time you launch PyMod, you will begin a new PyMod project. By beginning a new PyMod project, a directory named *new_pymod_project* will be created inside the *projects* directory in your **PyMod Directory**. As you start to import sequence and structures in PyMod, this directory will be populated with temporary and output files. You may find the following subfolders in any of your

project directories:

- *alignments*: it will contain alignments temporary input and output files.
- *images*: it will contain images you create using some of the functionalities of PyMod 2.
- *structures*: the coordinate files you import in PyMod 2 will be copied in this folder.
- *models*: PyMod 2 will place MODELLER output in this folder, creating a subfolder for each modeling session you perform in a project.
- *psipred*: it will contain PSIPRED output files.
- *similarity_searches*: it will contain (PSI-)BLAST output files.

4.4 Save and Load PyMod Projects

PyMod projects work in a way similar to PyMOL sessions. Once you have begun a new PyMod project, you may be able to save your progress by using the **File** → **Projects** → **Save Project as** option of PyMod main menu. In this way, you can save a file with a `.pmse` (PyMod session) file (a `.pmse` file is just a ZIP file storing a PyMOL session and other files containing all the PyMod data of your project). Once you have saved a `.pmse` on your disk, by using the **File** → **Projects** → **Open Project** command you may be able to load this file and restore PyMod and PyMOL to the state in which they were when you saved your project. PyMod projects are cross-platform (like PyMOL sessions) and they will run on operating systems different from the one on which they were built. However, old PyMOL versions will produce session files not compatible with recent PyMOL versions, and in this case PyMod projects will be incompatible too.

4.5 Input and Output

PyMod 2 can input and output some of the most popular protein sequence and structure file formats, so that protein sequences and structures may be imported, analyzed, edited and exported while using the plugin.

4.5.1 Input sequence files

4.5.1.A Open sequence files

PyMod can open protein sequences files in the commonly used FASTA and genpept formats. In order to load a protein sequence from these file types, use the **File** → **Sequences** → **Open from File** option in the plugin main menu. The sequences contained in the file will be loaded in the plugin main window.

4.5.1.B Input raw sequences

Using the **File** → **Sequences** → **Add Raw Sequence** option from the plugin main menu you will be

presented with window where you can insert the name and the amino acid sequence of a new element. The element will be loaded in PyMod once you press the editing window **Submit** button.

4.5.1.C Edit sequences

You may edit at any time the sequences you have loaded in PyMod. To do that, right-click on a sequences name in the left pane of the plugin's main window and use the **Sequence** → **Edit Sequence** option.

4.5.2 Input structure files

4.5.2.A Open structure files

In order to load a PDB file in PyMod, use the **File** → **Sequences** → **Open from File** option from the plugin main menu. Each polypeptide chain contained in the file will be split from the others, its sequence will be loaded as a separate element in PyMod main window and its 3D structure will be loaded in PyMOL as a separate object (see **Figure 5.5**).

4.5.2.B Import objects from PyMOL

If, when starting PyMod 2, you have already loaded in PyMOL some objects, you may import them in PyMod using the **File** → **Sequences** → **Import PyMOL Objects** command from the plugin main menu. A window asking you to select which PyMOL object to import will appear: select the objects you wish to import and then press the **Submit** button to load them in PyMod.

4.5.2.C Associate structures to sequences

You might use this feature in order to associate a 3D structure to a sequence loaded in PyMod lacking one. In order to use this function, right-click on a sequence name on the left pane of the plugin main window and then select the **Structure** → **Associate 3D Structure** option on the popup menu. You will be prompted with a window asking you to select a PDB file present on your system and the ID of one of its chains. If the sequence loaded in PyMod and the sequence of the polypeptidic chain you specified match, the 3D structure will become associated with the sequence loaded in PyMod (the association will occur even if the sequence loaded in PyMod matches only a part of the polypeptide chain in the structure file).

4.5.3 Input alignment files

4.5.3.A Open alignment files

PyMod 2 can open protein sequence alignment files in the fasta and clustal formats. In order to import an alignment in PyMod use the **File** → **Alignment** → **Open From File** option in the plugin's main menu. The sequences contained in the alignment file will be imported in PyMod and grouped in a new sequence cluster.

4.5.4 Output sequence files

4.5.4.A Save sequence files

You can save in the FASTA format any sequence (or sequence selection) by right-clicking on a sequence name on the plugin's main window's left pane and by selecting the *Sequence* → *Save Sequence to File* (or *Selection* → *Sequences* → *Save Selection to File* for selections) option.

4.5.4.B Save Alignment Files

You can save in the FASTA or Clustal format any multiple sequence alignment loaded in PyMod. To do that, go to the *Alignments* → *Name of the Alignment* → *Save to File* from PyMod main menu or alternatively right-click on the cluster name in PyMod main window left pane and use the *Edit Cluster* → *Save Alignment to File* option in the popup menu.

4.6 Analyzing Sequences and Structures

PyMod has a main window (see **Figure 4.4**) from which you can access all its functionalities.

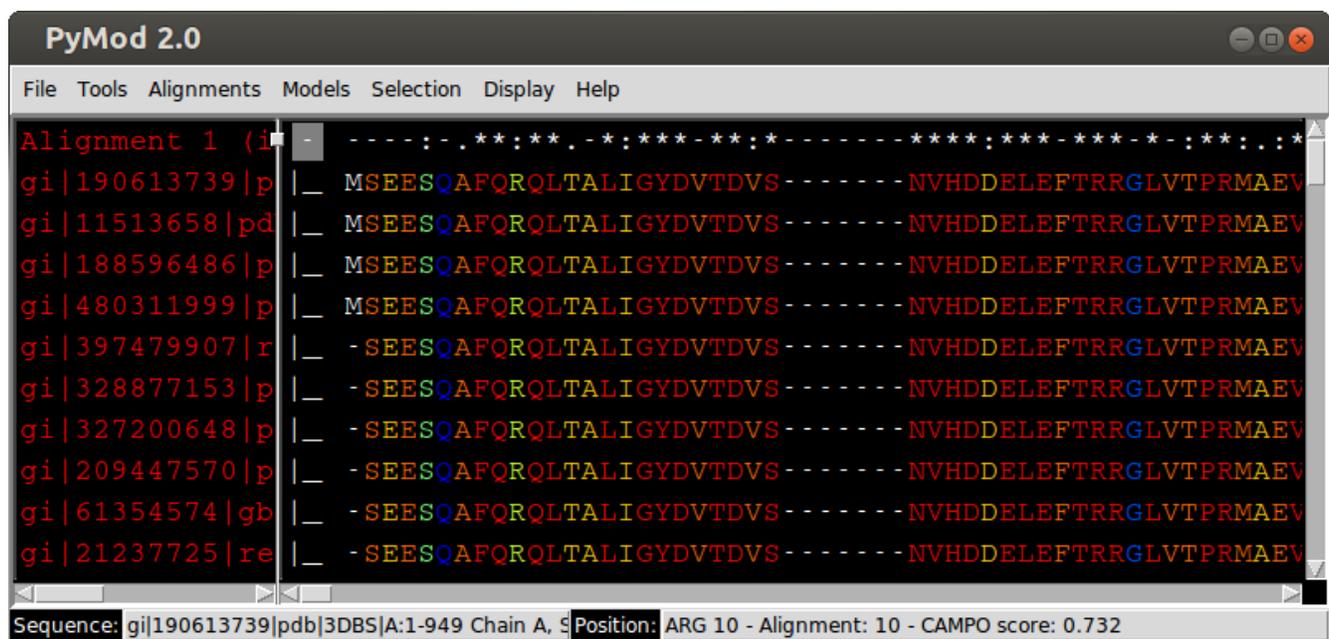


Figure 4.4: PyMod 2 main window.

4.6.1 PyMod main menu

The main menu of PyMod window contains the following submenus:

- **Files:** from there you may load new sequences, structures or sequence alignments files in PyMod/PyMOL.

- **Tools:** once you have loaded some sequences or structures in PyMod, you may select them in plugin main window and use one of the bioinformatics tools accessible from this submenu to analyze them.
- **Alignments:** once you have built a multiple sequence alignment in PyMod, you may analyze it through the functionalities accessible from this menu.
- **Models:** once you have built some homology models using MODELLER, you may analyze them using options from this menu.
- **Display:** here you can adjust PyMod main window font size and the color scheme of the sequences.
- **Help:** here you can access PyMod documentation.

4.6.2 Sequences

When you load a protein sequence or structure in PyMod (see **Input and Output**), its amino acid sequence will be loaded in PyMod main window right pane, while its name will be displayed in the main window left pane. You may interact with them using your mouse.

- **Left-click:** you can select sequences by left-clicking on their name: the color of their header will turn from red to green.
- **Right-click:** By right-clicking on a sequence name, you will open a popup menu with several options:
 - *Sequence Options:* from here you can edit, save or delete the sequences loaded in PyMod.
 - *Color Options:* from there you can color a sequence according to a series of color schemes available in the plugin. Available color schemes include for example by secondary structure (both observed and predicted), by DOPE (one of the MODELLER package energy functions) or by conservation (expressed using CAMPO score).
 - *Structure Options:* from here you can associate a 3D structure to sequences lacking one or you can manage the 3D structure of elements with associated 3D structures loaded in PyMOL.
 - *Cluster Options:* from here you can manage a sequence inside its cluster.
 - *Selection Options:* if you have selected multiple elements, from here you can manage all of them at the same time.

4.6.3 Structures

When you load in PyMod an element with an associated 3D structure (see **Input structure files**) its

sequence will be loaded in PyMod main window and its structure will be loaded in PyMOL.

Since you will probably spend most of the time in PyMod session analyzing sequences in the PyMod main window (performing operations such as the building, analysis and editing of target-template alignments), there are a series of shortcuts to allow you to switch from the analysis of protein sequence in PyMod main window to its corresponding 3D structure loaded in PyMOL.

- **Center a chain in PyMOL:** right-click on a structure name in PyMod main window and use the *Structure* → *Center Chain in PyMOL* option in the popup menu.
- **Show a chain in PyMOL:** right-click on a structure name in PyMod main window and use the *Structure* → *Show Chain in PyMOL* option in the popup menu.
- **Hide a chain in PyMOL:** right-click on a structure name in PyMod main window and use the *Structure* → *Hide Chain in PyMOL* option in the popup menu.
- **Quick showing and centering of a chain:** middle-click on a structure name in PyMod main window. If the sequence is in a “selected” state in PyMod main window (that is, if its header is colored in green) the chain will be centered in PyMOL. If the sequence is in a “unselected” state in PyMod main window (that is, if its header is colored in red) the chain will be hidden in PyMOL.
- **Quick centering and selection of a residue:** middle-click on a residue in PyMod main window. The residue will be selected and centered in the PyMOL viewer.

4.6.4 Alignments and clusters

Once you have built a cluster of sequences, you may edit the alignment by introducing or removing gaps in its sequences. Point the mouse on any residue of the alignment, then press the mouse left button and, while keeping the mouse button pressed, drag the residue towards the left (to remove gaps behind it) or right (to add gaps behind it).

4.7 Similarity Searches

Right now you can perform similarity searches in PyMod 2 by taking advantage of the BLAST and PSI-BLAST algorithms. You can use any sequence loaded in PyMod as a query and you may import in PyMod the homologous sequences identified in the search. If the hit sequences have a corresponding structure deposited on the PDB, this structure can optionally be imported in PyMod/PyMOL.

4.7.1 Using NCBI-BLAST

You can perform a BLAST search within PyMod by only having an internet connection to allow you to connect to the NCBI BLAST server. You may use any sequence loaded in PyMod as a query. In order to perform a search, select it by clicking on its name on the plugin main window left pane, then use the

Tools → *Database Search* → *BLAST* option on the plugin main menu.

The NCBI-BLAST preferences window will appear. There are several search options you may change through this window, according to the type of search you want to perform.

- **Database Selection:** this option will allow you to select the protein sequence database over which to perform the BLAST search. You can find a description of these databases at (<ftp://ftp.ncbi.nlm.nih.gov/blast/db/README>). If you are performing a similarity search to retrieve some templates for an homology modeling prediction, select the *PDB* database.
- **E-value Threshold:** the maximum E-value (see <https://www.youtube.com/watch?v=nO0wJgZRZJs> or <https://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml#expect> for a definition of E-value) of a match in order for it to be included in the search results.
- **Max Number of Hits:** the maximum number of hits to be reported in the search results.
- **Min ID% Threshold (Advanced Option):** the minimum sequence percentage identity between the query and the subject of the HSP in order for the match to be included in the search results.
- **Min Coverage% Threshold (Advanced Option):** the minimum span of the subject sequence in the HSP over the whole query sequence in order for the match to be included in the results.

4.7.2 Using PSI-BLAST

You must have installed PSI-BLAST and some sequences databases on your system in order to perform a PSI-BLAST search within PyMo 2 (for more information see the PSI-BLAST **Configuration within PyMod**). In order to perform a PSI-BLAST search, select any sequence loaded in the plugin by clicking on its name in the plugin main window left pane, then use the *Tools* → *Database Search* → *PSI-BLAST* option of the plugin main menu.

The PSI-BLAST preferences window shown in **Figure 4.5** will be displayed.

Figure 4.5. PSI-BLAST options window.

There are several options you might change from here.

- **Database Selection:** by using this option, you will be able to select the sequence database over which to perform the PSI-BLAST search. A list of sequence databases available on your system in the PyMod **BLAST Database Directory** will be displayed. If you wish to select another sequence database present on your system on a path different from the PyMod **BLAST Database Directory**, select the *Browse* option. If you wish to add a new sequence database to the list, see [Add a BLAST Sequence Database to PyMod 2](#).
- **PSI-BLAST Iterations:** the number n of iterations which the PSI-BLAST algorithm will perform. When you use PSI-BLAST within PyMod, all the matches identified in the n iterations will be displayed together in the results.
- **E-value Threshold:** see the same option in the [Using NCBI-BLAST](#) section.
- **Max Number of Hits:** see the same option in the [Using NCBI-BLAST](#) section.
- **Min ID% Threshold (Advanced Option):** see the same option in the [Using NCBI-BLAST](#) section.
- **Min Coverage% Threshold (Advanced Option):** The minimum span of the subject sequence in the HSP over the whole query sequence in order for the HSP to be included in the results.
- **PSI-BLAST E-Value Threshold (Advanced Option):** the E-value above which the subject of

the HSP will not be included by PSI-BLAST in the sequence profile that will be used in the next iteration to look up for homologous sequences.

4.7.3 Import hit sequences in PyMod

If your (PSI-)BLAST search has identified some hit sequences, you will be presented with a result window, where you can decide the list of the hit sequences to import in PyMod (see **Figure 4.6**).

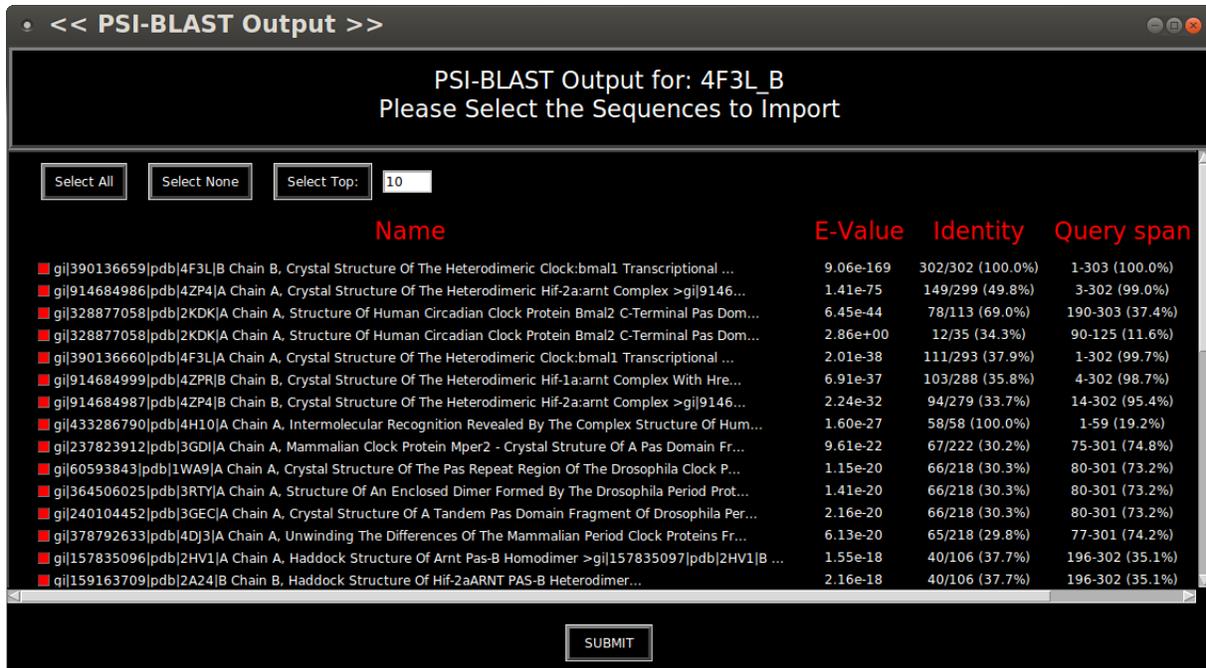


Figure 4.6. PSI-BLAST results window. From here you may select the hit sequences to import in PyMod.

Click on the hit sequences names to select them. Once you have selected the sequences to import, press the **SUBMIT** button of the result window in order to import them into PyMod.

Your hit sequence will be grouped in a new cluster together with the query sequence you used in the search. They will be aligned according to (PSI-)BLAST pseudo alignment (that is, each hit sequence will be aligned to the query sequence according to the pairwise alignment built by BLAST when performing the search). You may later rebuild the multiple alignment by using one of the multiple sequence alignment algorithms available in PyMod (see **Alignment Building**). Just select the entire cluster (by clicking on its name on PyMod main window left pane) and use one of the options available in the **Tools** → **Sequence Alignments** menu from PyMod main window.

4.7.4 Fetching structures over the PDB

If you have performed a search over the PDB sequence database and have imported in PyMod some hit sequences, you may also choose to import their 3D structures.

In order to fetch the 3D structures of an hit sequence over the PDB, right-click on its name in PyMod main window left pane and use the **Structure** → **Fetch PDB File** option of the popup menu.

You will be asked to choose among two different import modes. Here are their differences.

- **Import all chains:** by using this option you will import on PyMod/PyMOL all the content of the PDB file where the 3D coordinates of the hit sequence identified by (PSI)-BLAST are present. This is like opening from the **File** → **Sequence** → **Load from File** option of PyMod main menu the original PDB file. The hit sequence will be deleted from the query's cluster and it will be replaced by the chains present in the PDB file, which will be imported in PyMod/PyMOL and placed outside the query's cluster. The chain identified by (PSI)-BLAST will be colored with a vivid color, while all other chains present in the PDB file will be colored in gray. Since this option allows you to maintain all the polypeptide chains and heteroatoms present in the PDB file to import, you may want to use it if you intend to build a multiple chain homology model, or an homology model containing the template heteroatoms.
- **Import only the hit sequences fragments:** by using this import mode, you will import in PyMod/PyMOL only the 3D structure of the hit sequence fragment present in the HSP identified by (PSI-)BLAST. The hit sequence will remain inside the query's cluster and its structure will be loaded in PyMOL. Note that if you use this option, you will import only the 3D atomic coordinates of the polypeptide chain of the hit sequence fragment: no heteroatoms present in the original structure file will be loaded in PyMOL. Selecting this mode is useful when you are working with short query sequences comprising only one protein domain: by performing a (PSI-)BLAST search over the PDB, you may identify hit sequences having large multidomain structures. If you use this import option, you will be able to import in PyMod/PyMOL only the structure of the domain homologous to your query sequence.

4.7.5 Expanding clusters

If you perform a (PSI)-BLAST search using a query sequence already present inside a cluster, you will be presented with the following options:

- **Build a new cluster:** selecting this options will just extract the query sequence from its cluster. A new cluster comprising the query sequence and all the hit sequence which you decide to import into PyMod will be created.
- **Expand old cluster:** using this option you will keep your query sequence inside its existing cluster and all the new hit sequence identified in the search will be added to the same cluster [they will be aligned to the query sequence according to the pairwise alignments in the HSPs identified by (PSI)-BLAST].

4.8 Alignment Building

PyMod 2 includes support for several protein sequence and structural alignment tools. There are several ways in which you can use these tools within PyMod and in the following section they are going to be outlined.

4.8.1 Sequence alignments

In order to perform a pairwise or multiple sequence alignment within PyMod, select more than one sequence (by left clicking on their names in PyMod main window) and then select one of the sequence alignment algorithms in the *Tools* → *Sequence Alignments* submenu in the plugin main menu. The algorithms currently available in PyMod 2 are reported below.

- **ClustalW:** when using ClustalW within PyMod in order to align some sequences, an options window will appear. Here you can choose the scoring matrix, the gap open and extension penalties to be used in the alignment carried out by ClustalW.
- **Clustal Omega:** when using Clustal Omega within PyMod, you will be presented with an options window where you may type in extra command line options for running the program. As Clustal Omega default parameters are probably well suited for most kinds of alignments you might want to perform in PyMod: for this reason, if you are not an expert user you may leave the default parameters as they are.
- **MUSCLE:** by using this option, a sequence alignment using MUSCLE default parameters will be executed. Right now you can not change within PyMod MUSCLE default parameters.
- **SALIGN-seq:** when performing an alignment where at least one sequence has a 3D structure associated in PyMOL, you will be asked if you want the SALIGN algorithm to use the structural information in the dynamic programming algorithm in order to attempt to build a more accurate alignment.

Once you have performed a sequence alignment, the aligned sequences will be grouped in a new cluster.

4.8.2 Structural alignments

In order to perform a pairwise or multiple structural alignment within PyMod, select more than one structure (by left clicking on their names in PyMod main window) and then select one of the structural alignment algorithms in the *Tools* → *Structural Alignments* submenu in PyMod main menu. You may perform a structural alignment using the **CE-alignment** algorithm or a structural alignment algorithm implemented in the **SALIGN** module of MODELLER.

4.8.3 Profile alignments

Once you have built (or imported) one or more alignments, you might perform profile-to-sequence or (profile-to-profile) alignments by taking advantage of one among the profile-to-sequence (or profile-to-profile) alignment algorithms implemented in ClustalW, Clustal Omega and SALIGN.

- **Profile-to-sequence alignment:** in order to perform a profile-to-sequence alignment, select an entire cluster (by left-clicking on its name on PyMod's main window's left pane) and at least one sequence outside the cluster. Then use the *Tools* → *Profile Alignments* submenu of the plugin main menu to select an algorithm to perform the profile-to-sequence alignment. In this way the algorithm will build a sequence profile out of the the cluster you selected and it will align the external sequence to it. Once the alignment has been performed, these external sequence will become part of the cluster they were aligned to.
- **Profile-to-profile alignment:** to perform a profile-to-profile alignment, select two entire cluster (by left-clicking on their name on PyMod's main window's left pane) and use the *Tools* → *Profile Alignments* submenu on the plugin main menu to select an algorithm to perform the profile-to-profile alinement. Once the alignment has been performed the two aligned clusters will be joined in a single cluster with all their sequences aligned according to the profile-to-profile alignment.

4.8.4 Additional alignment modes

PyMod provides two additional alignment modes, which you might find useful in several kinds of occasions. Here is a description of them and of how to perform them.

4.8.4.A Keep previous alignment mode

This mode will be available whenever you select one or more sequences inside a target cluster, and one or more external sequence which do not belong to any other cluster: when performing an alignment using a tool from the *Tools* → *Sequence Alignments* or *Tools* → *Structural Alignments* you will be asked to choose among two alignment modes.

- **Build a new alignment :** if you select this mode you will just build a new alignment containing the sequences you selected.
- **Keep the previous alignment:** if you select this option, you can perform this particular alignment mode. In this mode:
 1. The external sequences will first be aligned together with sequences you selected in the target cluster in a multiple sequence alignment using the algorithm you chose in the main menu (if you chose to a structural alignment, they will be aligned with ClustalW by default).
 2. Once this initial multiple alignment has been built, PyMod will find for each external

sequence the most similar sequence inside the target cluster (using sequence percentage identities as a measure of sequence similarity).

3. Each external sequence will be independently pairwise-aligned to its target sequence by means of the algorithm you originally selected. The external sequences will then be appended to the target cluster by merging their pairwise alignment with their target sequence with the rest of the alignment.

4.8.4.B Alignment joining mode

This mode will be available whenever you select a single sequence for a series of two or more clusters use and use an algorithm accessible from the *Tools* → *Sequence Alignments* or *Tools* → *Structural Alignments* menus. The sequences you selected will be used as bridges to join their corresponding alignments. That is, in this mode:

1. PyMod will first execute an alignment (using the algorithm you selected) using the bridges sequence (or structures).
2. Then, it will use the alignment between the bridges in order to merge all the clusters by keeping in frame the bridges with the other sequences in their clusters.

This mode is particularly useful when you have two (or more) remotely related homologous sequences with a 3D structure loaded in PyMOL and each sequence is aligned to a cluster of closely related homologous sequences. You might use the sequences with the 3D structure as bridges and perform a structural alignment: in this way you can merge their corresponding clusters of distantly related sequences to yield an accurate alignment (guided by structural information) that would have been difficult to obtain otherwise.

4.9 Alignment Analysis

4.9.1 Sequence clusters and alignments

In PyMod 2, every time you either :

- Build a new alignment from scratch
- Perform a (PSI)-BLAST search and import a set of hit sequences
- Import an alignment from a FASTA or Clustal file

a new sequence cluster will be created. Sequence clusters can be collapsed or expanded by left-clicking on the + or - buttons beside them in PyMod main window right pane (see **Figure 4.7**).

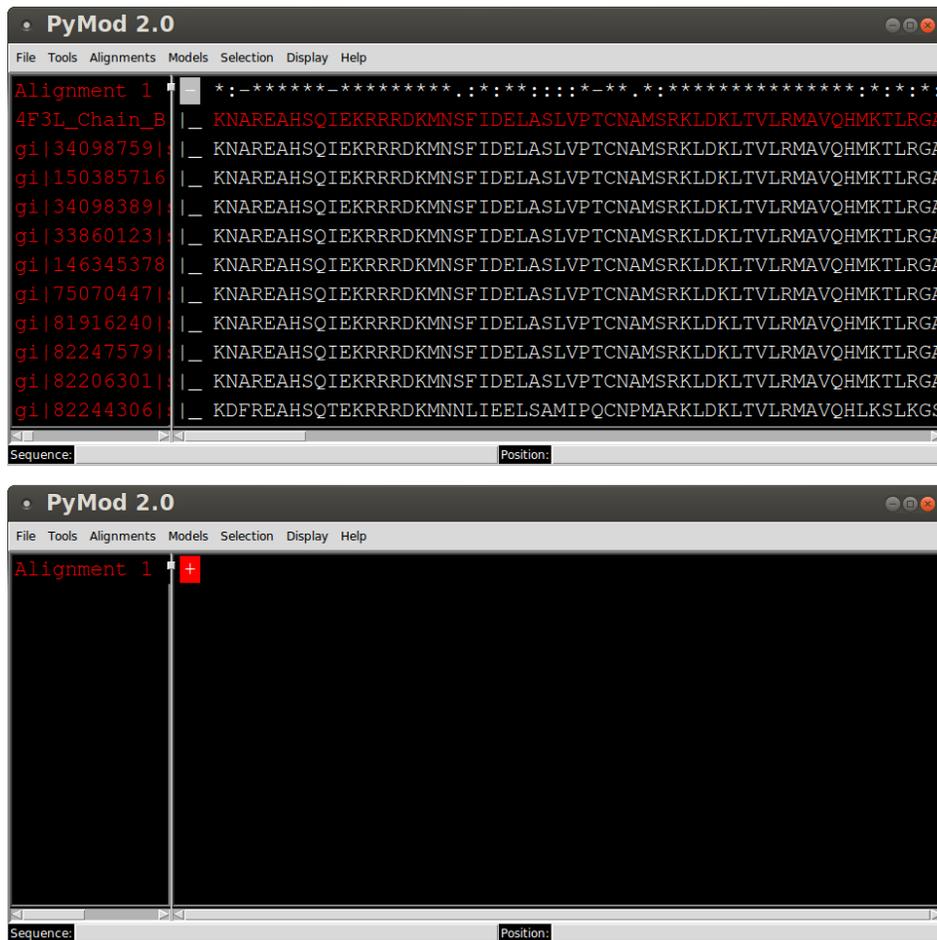


Figure 4.7. In the upper image, a cluster of sequences in the PyMod main window is shown in its expanded state. In the lower panel, the cluster is shown in its collapsed state. In order to switch between the two states, press the + red button (to expand a collapsed cluster) or the – gray button (to collapse and expanded cluster).

For each sequence cluster present in PyMod at any time, an item in the *Alignments* menu on PyMod main menu bar will be created. For each cluster you may perform several operations using the options accessible from this menu.

4.9.2 Matrices menu

For each cluster a matrices submenu is available. From here you can access the following functionalities.

- **Identity Matrix:** using this command you can display a table containing the sequence percentage identity matrix computed from the current sequence alignment contained in the cluster.

- **RMSD Matrix (for structural alignments only):** each time you build a structural alignment using either CE-alignment or SALIGN-str, using this option you can access the RMSD matrix of the structural alignment. The RMSD is computed between all pairs of alpha carbon of the equivalent residues in the sequence alignment generated from the structural alignment program.

4.9.3 Trees menu

From this menu, you can access a series of options depending on the type of alignment you built.

4.9.3.A Show Guide Tree

Whenever you perform a multiple sequence alignment using either ClustalW, Clustal Omega (which utilize guide trees to carry out multiple progressive sequence alignments) you have the chance to display the tree using the *Alignments* → *Alignment n* → *Trees* → *Show Guide Tree* command of the alignment submenu.

4.9.3.B Build Tree

By using the *Alignments* → *Alignment n* → *Tree* → *Build Tree from Alignment* option, you may build *de novo* a distance tree out of a multiple sequence alignment. In order to use this option you need to have the ClustalW package installed on your machine (see **ClustalW**) as this package implements algorithms to build distance trees out of multiple sequence alignments. If you use this command, you will be presented with a window with the following options.

- **Clustering Algorithm:** from here you can decide to use either the Neighbour Joining (https://en.wikipedia.org/wiki/Neighbor_joining) or the UPGMA (<https://en.wikipedia.org/wiki/UPGMA>) algorithms to build your tree.
- **Use Distance Correction:** by selecting this option, ClustalW will attempt to correct for multiple substitutions at the same site. This is recommended for more divergent sequences and has the effect of stretching branch lengths (however, for very divergent sequences the distances cannot be reliably corrected).
- **Exclude Gaps:** by selecting this option, the columns where any of the sequences in the input have a gap will be excluded, forcing ClustalW to use only positions where information can be included from all sequences.

When you press the **Submit** button on the options window, ClustalW will build a distance tree of the alignment using the parameters you selected and the tree will be displayed using the Phylo [Talevich et al., 2012].

4.9.4 Evolutionary conservation menu

4.9.4.A CAMPO

Using the *Alignments* → *Alignment n* → *Evolutionary Conservation* → *CAMPO* command you can run the CAMPO algorithm [Paiardini et al., 2005] (see **CAMPO**) to analyze the multiple sequence alignment contained in any cluster present in PyMod. When you decide to run CAMPO, you will be presented with a window where you can modify the following parameters of the algorithm.

- **Scoring Matrix Selection:** in order to assess the degree of evolutionary conservation in a multiple sequence alignment, CAMPO will compute a series of substitution scores for any two pairs of residues occurring in a column of multiple sequence alignment. By changing this option you may decide which matrix is going to be used to assign these scores.
- **Gap Score:** the score CAMPO will assign for residues-to-gap pairs occurring in a column of a multiple sequence alignment.
- **Gap-to-Gap Score:** the value CAMPO will assign for gap-to-gap pairs occurring in a column of a multiple sequence alignment.
- **Toss Gaps:** if you select this option, CAMPO will not assign a conservation score to columns of a multiple alignment in which more than 75% of the positions are occupied by gaps.

When you press the *Submit* button PyMod will run CAMPO and the sequences (along with their respective structure loaded in PyMOL) of the alignment will be colored according to their CAMPO scores, with red indicating an higher CAMPO score (that is, an higher level of evolutionary conservation) and blue indicating a lower CAMPO score (that is, a lower level of evolutionary conservation) .

4.9.5 Alignment rendering menu

PyMod implements a quick access to online tools for alignment analysis and rendering. You may submit to the following tools the multiple sequence alignment contained in any cluster present in PyMod.

4.9.5.A WebLogo 3

By using the *Alignments* → *Alignment n* → *Render Alignment* → *Generate Logo through WebLogo 3* you may build a sequence logo of a multiple alignment by means of the WebLogo 3 [Crooks et al., 2004] web tool. When you use this command, you will be presented with a window where you can change the parameters which the WebLogo 3 server will use to build the sequence logo. Since these options are essentially the same present in the WebLogo 3 web interface (<http://weblogo.threeplusone.com/create.cgi>) please refer to the site's documentation if you want more information about them.

Once you press the **Submit** button, PyMod will send a request to the Weblogo 3 server, retrieve the logo output and show it using our default pdf or image viewer (depending on the **Logo Format** option you chose). The file contained the logo will be saved inside the `./images` directory of your current project directory.

4.9.5.B ESPrpt

By using the **Alignments** → **Alingment n** → **Render Alignment** → **Launch ESPrpt in Web Browser** command you may decide to use to render a multiple alignment by means of the ESPrpt web tool [Robert and Gouet, 2014]. When you use this command, an alignment file contained the aligned sequences in your target cluster will be uploaded on ESPrpt's server and your default browser will open on ESPrpt editing window (<http://esprpt.ibcp.fr/ESPrpt/cgi-bin/ESPrpt.cgi>). Here you can proceed to change ESPrpt parameters or render the alignment using the **SUBMIT** button on the top-left corner of the web page layout.

When using this functionality of PyMod 2, if some sequence in the target cluster have a 3D structure associated in PyMOL, you will be presented with a window asking you to select one of the structures. By selecting a structure, its secondary structure content will be represented in image generated by ESPrpt.

4.10 Sequence and Structure Analysis

4.10.1 PSIPRED

If you have installed PSIPRED on your system (see **PSIPRED**) you may it within PyMod to predict the secondary structure content of any sequence loaded in the plugin. In order to run PSIPRED within PyMod, select a sequence in the plugin main window and use the **Tools** → **Structural analysis** → **PSIPRED** option. PSIPRED predictions may take a while, depending on the sequence length (on a modern desktop computer, it will take approximately 5-10 seconds for a 200 residue protein). Once PSIPRED has completed its prediction, the target sequence's residues will be colored with the following color scheme :

- **Red:** predicted to be in an alpha helical conformation .
- **Yellow:** predicted to be in a beta sheet conformation .
- **Green:** predicted to be in an aperiodic conformation .

Each residue will be colored with a different shade, according to the confidence value assigned to it by PSIPRED. Darker shades correspond to higher confidence. Lighter shades to lower confidence. You may read the confidence level of the prediction for any residue by hovering with your mouse on that residue in PyMod main window: the value will be displayed in the **Position** messagebar in PyMod main window (see **Figure 4.8**).

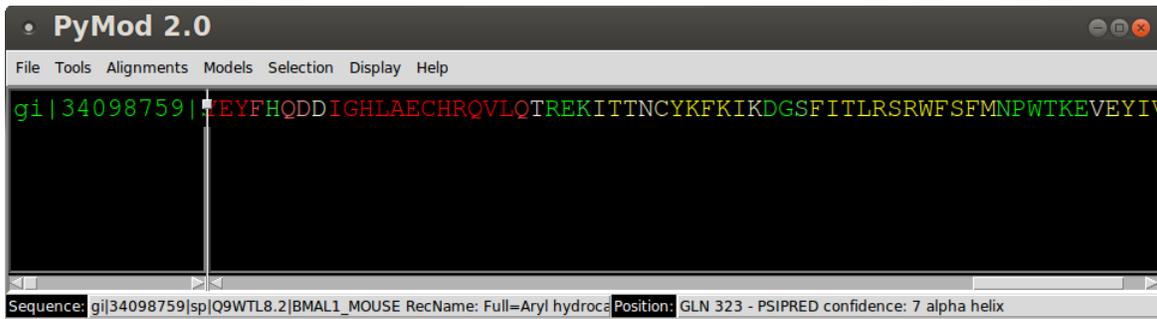


Figure 4.8. When using PSIPRED, the input sequence will be colored according to the PSIPRED prediction. The confidence values for each residue are shown in the bottom right of the window while hovering with the mouse over them.

4.10.2 Ramachandran plot

When you have loaded a structure in PyMod/PyMOL, you may build its Ramachandran plot (see **Figure 4.9**) by using the *Tools* → *Structural Analysis* → *Assess DOPE* command in PyMod main menu. When using this functionality, you may decide to draw in the plot all the 20 different amino acids, or only a subset of them. By middle clicking on the points you may select and center the corresponding residues in PyMOL.

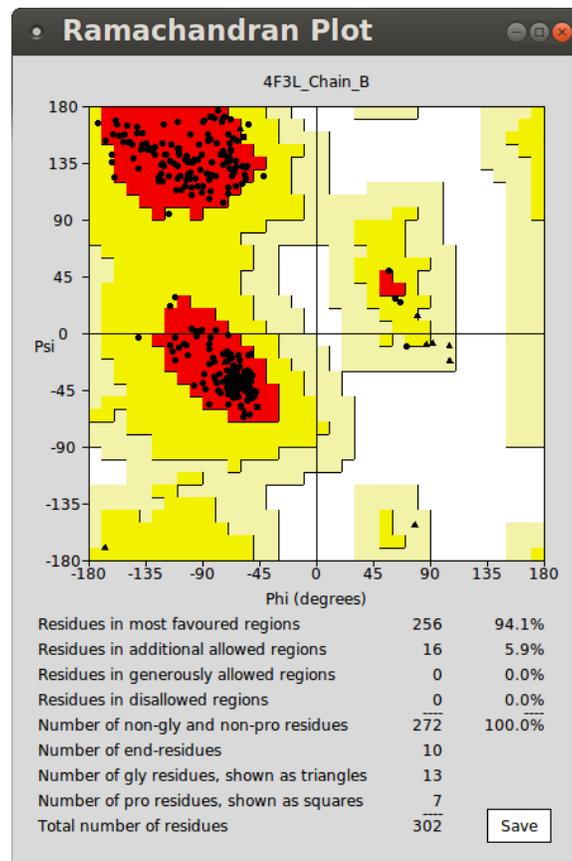


Figure 4.9. PyMod 2 interactive Ramachandran plot.

4.10.3 Assess DOPE

If you have MODELLER installed on your system, you may compute the DOPE [Shen and Sali, 2006] of any protein structure loaded in PyMod/PyMOL. Computing the DOPE is useful when assessing the quality of theoretical models of protein structures. You may want to compare the DOPE profiles (that is, the DOPE energies of each residue in a protein) of a homology model with that of its templates (or also compare multiple versions of an homology model), in order to highlight their local differences.

To compute the DOPE energy of one or more protein structure loaded in PyMod/PyMOL, select them in PyMod main window and use the *Tools* → *Structural Analysis* → *Assess DOPE* option from PyMod main menu. You will be presented with a plot with the DOPE profiles of the structures you selected. Hover with the mouse on the profiles lines and left-click on them to select and center the corresponding residue in PyMOL.

4.11 Homology Modeling Using MODELLER

If you have MODELLER installed on your system, you may use PyMod 2 as graphical user interface to

build homology models of protein structures. In the following section we will illustrate how to use MODELLER within the PyMod/PyMOL environment.

4.11.1 Building homology models

Once you have generated a cluster in which a target sequence lacking a structure is aligned to one or more templates (in PyMod any sequence with an associated 3D structure loaded in PyMOL can be used as a template), you may run MDOELLER to build an homology model of your target sequence utilizing the target-template(s) alignment you prepared. There are two homology modeling modes in PyMod 2: **single chain modeling** mode and **multiple chains modeling** mode. In the two sections below, we will illustrate how to build a model in both modes.

4.11.2 Single chain homology modeling

You can perform this mode when you have a single target sequence (that is, a single target protein chain) of which you would like to build an homology model. In order to launch this mode, make sure that your target sequence is currently aligned to at least one sequence having a 3D structure loaded in PyMOL (that is, a template). Select the target sequence by left-clicking on its name in PyMod main window and then use the *Tools* → *Homology Modeling* → *MODELLER* command of the plugin main menu. You will be presented with PyMod **Homology Modeling Window**. In this window you can set several parameters of the modeling process.

4.11.3 Multiple chain homology modeling

You can perform this mode when you have more than one target sequence (that is, more than one target protein chain) and you want to build a model of the complex of the target chains. This mode can be accessed whenever you have identified a **Template Complex**, that is, a structure file containing multiple chains homologous to your target chains. To build a multiple chain homology model within PyMod, each target chain must be aligned in a separate cluster to a different **Target Complex** chain. In other words, if you have a **Template Complex** consisting of n chains and you have n target sequences, you will have to align independently each of your target chains to a corresponding chain of the template complex, thus obtaining n different clusters in which one of your target chains is aligned to a different **Template Complex** chain. You may also build a multiple chain model using only a subset of the **Template Complex** chains. For example you may decide to build a model consisting of two target chains using a **Template Complex** having four chains: you just have to build two clusters having each a target chain and a corresponding **Template Complex** chain.

In the model building phase, MODELLER will build a 3D model of the target complex by restraining the target chains relative positions and orientation using the quaternary structure information contained in the **Template complex**. While you can use only one **Template Complex** per modeling session, you may define for each of your target chains additional templates chains. In order to do that, you will have

to including them in the alignments comprising your target sequences a **Template Complex** chains. In the model building phase, MODELLER will use the additional structural information deriving from these extra templates to model the target chains they were aligned to.

Once you have built your target-template alignments, in order to launch the **Modeling Window** of this modeling mode, select all your target sequence by left-clicking on their names in PyMod main window and then use the *Tools* → *Homology Modeling* → *MODELLER* option of PyMod main menu.

4.11.4 The Homology Modeling Window

Whether you perform PyMod single chain or multiple chain homology modeling mode, you will access the same **Homology Modeling Window**, where you can set several parameters of the model building phase. The modeling window is organized in three tabs, described below:

- **Main tab:** here you can select which template to use in the modeling process and what kind of structural information to include in your models (such as templates heteroatoms, water molecules).
- **Disulfide tab:** here you can instruct MODELLER on how to build the disulfide bridges of your models.
- **Options tab:** here you can define some important parameters of the modeling process.

Once you have set all the parameters according to your needs, you can launch MODELLER by pressing the *Submit* button on the **Homology Modeling Window**.

In the following sections the options that you can access from the **Homology Modeling Window** are going to be described.

4.11.5 Main tab

In this tab you can select which templates to use in the modeling process and other modeling options. You will find a description of the options you may modify in this tab in the sections below.

4.11.6 Template complex selection in the Main tab (multiple chain homology modeling mode only)

When PyMod identifies from the target-template alignments you built more than one suitable **Template Complex**, you can define which one to use in this section of the **Main** tab. Multiple Template Complex may be identified in cases like the one shown as an example in **Figure 4.10**.

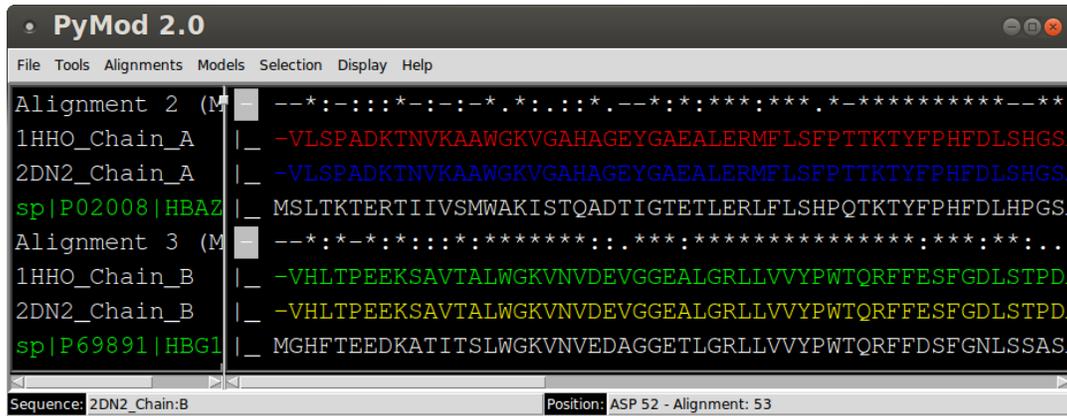


Figure 4.10. Here there are two clusters in which a target sequence (which names are colored in green) is aligned to a different chain originating from the same PDB file: in this case the both the 1HHO and 2DN2 structures could potentially be used as **Template Complexes**.

If you choose one template complex over the others, you can still use their chains as additional templates to model the single chains of your multiple chain model.

4.11.7 Modeling options in the Main tab

For each of your target sequences, you will have a **Modeling Options** section. In this part of the **Main** tab of the **Homology Modeling Window**, you will find the following sections.

4.11.7.A Restraints options

- **Use symmetry restraints for this chain (multiple chain modeling mode only):** if you are building a model with multiple chains and you have at least two chains sharing the exact same sequence (this can happen when you are building models of proteins with multiple copies of the same subunit, that is of any kind of homomer), you may decide to instruct MODELLER to apply symmetry restraints for these chains. In this way, in the model building phase, MODELLER will try to restrain the chains conformations in order for them to be as similar as possible. This is suitable when building symmetric protein complexes, because by default MODELLER will probably generate slightly different conformations for each subunit.

4.11.8 Template selection in the Main tab

For each of your target chains, you will be presented with a list of “boxes”, each corresponding to a template. In this part of the **Main** tab of the **Homology Modeling Window**, you can select which templates you will be using in the model building phase. Here is a list of the options you will find inside each template chain box.

4.11.8.A Use as Template

Here you will find the name of the templates aligned to your target sequence and in brackets you will find their sequence percentage identity with the target chain they are aligned to. Click on this check button beside the template name if you want to use this template in the model building phase. MODELLER will only use templates you select using this option.

4.11.8.B Hetero Residues

If a template chain has any non-water hetero residues in its structure file, you may specify which one to use in this section. You can choose among three different options.

- **Use all heteroatomic residues:** by selecting this option all the hetero residues (of which the total number is indicated in brackets) of the template chain will be included in your model.
- **Select single heteroatomic residues:** by selecting this option, you will be presented with a list of all the non-water hetero-residues of the template chain. They will be indicated with their three letter code, their residue ID in the template original PDB file and in brackets it will be reported if the hetero-residue is a ligand or a modified residue. By clicking on the hetero residues names you may select which one to include in your model.
- **Do not use any heteroatomic residue:** by selecting this option no hetero residue belonging to this template chain will be included in your model.

4.11.8.C Include Water

If template chain has any water molecules in its original structure file, check this option to let MODELLER include them in your model. If you have multiple templates currently aligned to your target sequence you may include in your model the water molecules of only one template: this restriction is present in PyMod to prevent building of poor quality models with overlapping water molecules coming from multiple templates.

4.11.9 Disulfides tab

MODELLER has support for several disulfide bridges building options. If one of your target sequence has at least two cysteine residues (that is, the minimum number of residues needed to form at least one disulfide bridge), in this tab you may select among three different modes for disulfide bridges building. You may use any of this modes by clicking on *Yes* or *No* radiobutton besides their names.

4.11.9.A Use templates disulfides

This mode will be selected by default (as it represents MODELLER default behavior) whenever one of your templates has some disulfide bridges. By choosing this option, if the cysteine residues engaged in a disulfide bridge in the template are conserved in the target sequence (according to the target-template alignment you built), MODELLER will build an equivalent disulfide bridge in the model. By clicking on the *Show* button, a list of all the disulfide bridges found in the templates will be shown. This list

reports the cysteine residues (and their PDB residue index) forming the template bridges and the corresponding residues in the target sequence. If the two corresponding residues in the target are cysteines, the element in the list will be colored in white, otherwise it will be colored in gray.

4.11.9.B Create new disulfides

Through this option you may define a series of cysteine residues pairs from the list of all cysteine residues in each of your target sequence. If you select this option, two comboboxes for selecting a cysteine pair will appear. When you select two cysteine residues, you may press the **Add** button in order to add that pair of cysteines to the custom disulfide bridges list. If these two cysteine residues are going to be placed sufficiently close in the final model (that is, their S atoms must have a distance of less than 2.5 Å), MODELLER will generate a disulfide bridge between them. This option may be utilized along the *Use template disulfides one*: in this way you can build a model containing both disulfide bridges from its template and additional ones defined using this option.

4.11.9.C Automatically build disulfides

By selecting this option, MODELLER will automatically build a disulfide bridge among all those cysteine residues finding themselves close enough in the model (that is, with their S atoms having a distance of less than 2.5 Å). By using this option, the two options above will be disabled. NOTE: try avoid using this option for target sequence having numerous cysteine residues, as in this situation a single cysteine residue could potentially be engaged with more than one cysteine partner, thus generating an ambiguity that will not let MODELLER proceed to build the model.

4.11.9.D How to build models without disulfide bridges

In order to build a model without any disulfide bridges (that is, with all its residues in a “reduced” state) just set to *No* value each of the previous options.

4.11.10 Options tab

In this tab you will find a series of options allowing you to change some important parameters of the model building phase and of the model analysis phase.

4.11.10.A Models to Build

From here you can define the number of models that MODELLER has to build (the maximum allowed number is PyMod is 1000). Building multiple models of the same target is usually a good idea, as structurally divergent regions in the model can be regarded as being probably the ones modeled with low accuracy compared with the rest of the model.

4.11.10.B Exclude Heteroatoms

By setting this option value to *Yes*, you will exclude all the template heteroatoms (both water and non water ones) from the modeling process, and all the options related to heteroresidues in the **Main** tab of

the **Homology Modeling Window** will become disabled.

4.11.10.C *Optimization Level*

Here you can define the level of optimization carried out by MODELLER in the model building phase. MODELLER builds its models by performing an optimization and refinement process (see <https://salilab.org/modeller/manual/node11.html> and the *Optimize the model* section in <https://salilab.org/modeller/manual/node493.html>). Briefly, the objective function of a model is first optimized with the variable target function method (VTFM) with conjugate gradients, and then the model is refined using molecular dynamics (MD) with simulated annealing. By changing the values of this option, you may define the level of optimization and refinement that MODELLER has to carry out in the model building phase. Generally, the accuracy of models is likely to increase by performing more refinement during the model building phase of MODELLER, but that comes at a cost of increased computation times. You can select among different levels of optimization and refinement.

- **Low:** MODELLER will use a lower level of VTFM optimization with respect to its default behavior.
- **Default:** the default level of optimization used by MODELLER `automodel` class. This value is the one selected by default.
- **Mid:** MODELLER will use a slightly more thorough VTFM optimization and MD refinement with respect to its default behavior.
- **High:** MODELLER will use a much more thorough VTFM optimization and MD refinement with respect to its default behavior.

4.11.10.D *Additional Energy Minimization*

By using this option you may use MODELLER in order to optimize the stereochemistry and non-bonded contacts of your models by carrying out an additional energy minimization by using the CHARMM 22 force field. This additional energy minimization will occur after the models are built and therefore after MODELLER has already performed the optimization and refinement schedule defined in the *Optimization Level* option of the modeling window (see above).

By default no additional energy minimization will be performed. If you decide to use this feature, you will be shown with a series of options allowing you to define the parameters of the minimization process (see **Figure 4.11**). Here is a description of these options

- **Perform Additional Cycles of Energy Minimization:** here you may select the series minimization algorithms you want to use and the number n of minimization cycles you want to perform using each algorithm. Only if you click on the *Perform* checkbox, the n cycles of minimization using a certain algorithm will be executed. The algorithms available are the following

- **Steepest Descent:** 80 cycles of steepest descent minimization will be carried out by default.
- **Conjugate Gradients:** 20 cycles of conjugate gradients will be carried out by default.
- **Quasi Newton:** no quasi Newton optimization cycles will be carried out by default.
- **Molecular Dynamics:** perform n cycles of molecular dynamics optimization at a fixed temperature (you may define its value in Kelvins in an apposite entry). By default, no molecular dynamics cycles will be performed.

The order with which each series of cycles will be executed is from top to down of the list in which they appear (with the Steepest Descent executed first, the Conjugate Gradients after, then the Quasi newton and finally the Molecular Dynamics cycles).

- **Select Components to Minimize:** the additional energy minimization will be carried out only the types of interactions you define in this section.
 - **Bonds:** minimize covalent bonds energies.
 - **Angles:** minimize covalent angles energies.
 - **Dihedrals:** minimize covalent dihedral angle energies.
 - **Impropers:** minimize improper dihedral angle energies.
 - **Lennard-Jones:** minimize Lennard-Jones energies.
 - **Coulomb:** minimize electrostatic interactions energies.
 - **Non Bonded Cutoff (Å):** the maximum distance (in Å) over which non-bonded interactions (that is, Lennard-Jones and electrostatic interactions) between two atoms will not be computed. The default value is 4 Å.

Please note that performing too much energy minimization by using this option, will most likely result in the a decrease of the accuracy of your models, therefore you should use the *Additional Energy Minimization* feature with caution.

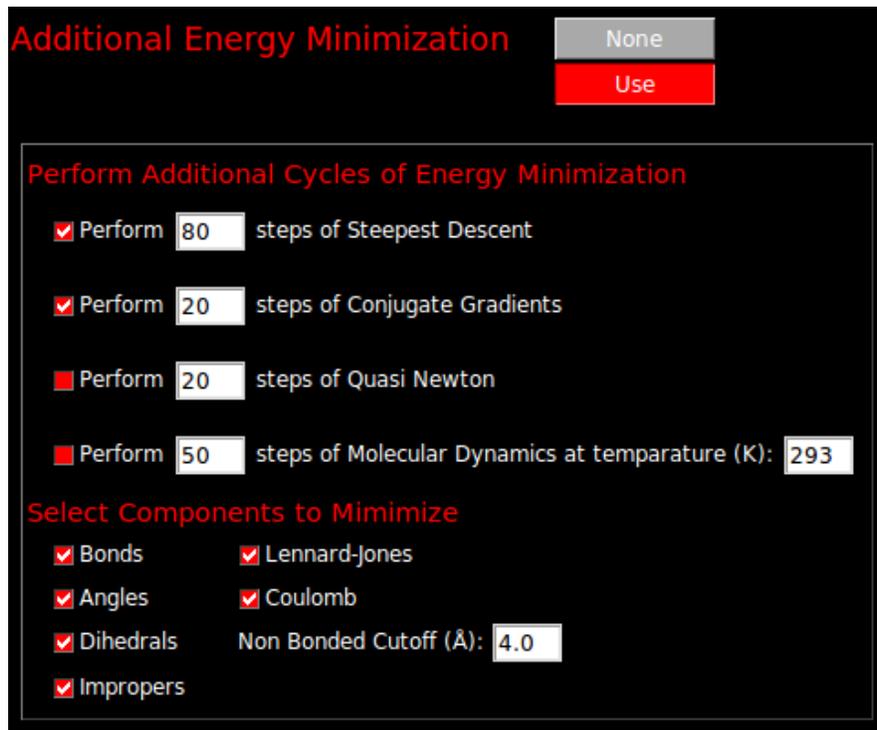


Figure 4.11. Additional energy minimizations options.

4.11.10.E Colors Models By

By selecting the *Default* value the 3D structure of the models loaded in PyMOL will be colored uniformly using PyMod color palette. By selecting the *DOPE values* option each residue of the models will be colored according to its DOPE value as computed by MODELLER, with lower energies tending to blue, while higher energies tending to red.

4.11.10.F Superpose to Models to Templates

By setting this option value to *Yes*, once the models are built and their 3D structure loaded in PyMOL, they will be superposed to the structures of their templates.

4.11.11 Analyzing results

When you have selected your templates and have chosen the modeling parameters, you can run MODELLER by pressing the *Submit* button on the **Homology Modeling Window**.

MODELLER will take some time before completing the model building phase, depending on the complexity of the system you chose to model. For example building five copies of a single chained ~200 residue target without performing any optimization (that is, with the *Optimization Level* option set to *None*) will take ~5 minutes on an modern average personal computer. Computation times will increase with multiple chain modes and with bigger proteins.

Once MODELLER has finished the model building phase its output will be loaded in PyMod/PyMOL. At the end of a modeling session:

- A table showing some assessment values for each model will be shown.
- A plot with the DOPE profiles of the models and templates will be shown.
- The 3D structures of the models will be loaded in PyMod/PyMOL.

4.11.11.A *Assessment Table*

After a modeling session, a table containing the list of all the models you built in the session will be shown. For each model the total objective function value and its total DOPE score are reported. Since lower objective function and DOPE score values tend to correspond to higher model quality, you may inspect these list of values in order to choose the model with the most favorable value. For multiple chain models, the values of the functions corresponds to the value computed for the whole complex. If you close the window containing this table, you may display it again by using the *Models* → *Modeling Session n* → *Assessment Table* option in PyMod main menu.

4.11.11.B *DOPE Plots*

After a modeling session, a plot containing the DOPE profile of all your models and templates will be shown. In PyMod DOPE profiles report the DOPE of all the standard amino acid residues of the structures. To facilitate comparison between the templates and the target profiles, the same gaps present in the target-template alignment you prepared will be introduced in the profiles, so that equivalent residues of the target and templates will also be “aligned” into the plot. Regions in models profiles presenting drastically higher DOPE values compared with the corresponding region in the templates, probably highlight local errors in the models.

In order to map on the 3D structure a certain region the DOPE profile, hover that region of the profile with the mouse and left-click on it to select in PyMOL the corresponding residue.

When building multiple chain models, DOPE profiles of the model chains (and their corresponding templates) will be displayed one after the other, resulting in long profiles. You may use the controls available at the bottom of the plot window to zoom on any region of the plot.

If you close the window containing the DOPE plot, you may display this plot again using the *Models* → *Modeling Session n* → *DOPE Profile*.

4.11.11.C *Model structures in PyMOL*

Once MODELLER has built the models, they will be loaded in PyMod/PyMOL. In any PyMod session, when you build the first model of a target sequence, no new element will be created in the main window: the existing target element in PyMod main window will just become associated with the new structure loaded in PyMOL. If you build additional models of the same target sequence within the same PyMod session, new copies of the target sequence associated with the new elements will be

created in the PyMod main window.

You may inspect your models 3D structure directly in PyMOL. Multiple chain models will be split as usual in separate chains.

If you want to save on your system a model coordinate file (in the PDB format) use the *Models* → *Modeling Session n* → *target_name.99990001* → *Save to File* option. In case of multiple chain models, a file containing all the models chains will be saved. You can also retrieve the models structure files built by MODELLER in a modeling session by looking in the `./projects/your_project_name/models/model_n_target_name` folder in your **PyMod Directory**.

4.12 PyMod 2 Options

By using the *Tools* → *Options* command in PyMod 2 main menu, you will be prompted with a window in which you can set several options of the plugin such as the path of the **PyMod Directory**, the path of the external tools executable and data files, **BLAST Database Directory** and MODELLER execution mode.

4.13 BLAST Sequence Databases

PyMod 2 offers an interface to the BLAST+ suite PSI-BLAST program (see **PSI-BLAST**). This program is executed locally and in order to use it, you will need some sequence database over which to perform a search. If you have used the `install_all.py` script of the **PyMod Installer Bundle**, the SwissProt and PDB sequence databases will be already present on your system and PyMod 2 will be already configured so that PSI-BLAST may use them.

If you want to install additional sequence databases or if you did not use the **PyMod Installer Bundle** to obtain the SwissProt and PDB databases, in the following section you will find instructions on how to add a BLAST sequence database to PyMod.

4.13.1 Add a BLAST Sequence Database to PyMod 2

- In this example we will show how to manually add to PyMod 2 the SwissProt sequence database.
- Protein sequence databases readable by the BLAST+ suite can be downloaded from: <ftp://ftp.ncbi.nlm.nih.gov/blast/db/>. You can download the SwissProt one from <ftp://ftp.ncbi.nlm.nih.gov/blast/db/swissprot.tar.gz>.
- Once you have downloaded it, unpack its content into a new folder named *swissprot*. This folder should contain all the files present in the `.tar.gz` file you unpacked (such as *swissprot.00.phr*, *swissprot.00.pin*, etc...). Note that if you download other databases, the name of the folder containing all their files should have the same prefix of their names (for

example the PDB sequence database is composed of files named *pdbaa.00.phr*, *pdbaa.00.pin*, etc... and the directory containing its files must be called *pdbaa*).

- Move the *swissprot* folder inside your **BLAST Database Directory** (by default the *data/blast_databases* folder of your **PyMod Directory**). Now the database should be accessible to PSI-BLAST (its name should appear in the list of available databases when you want to perform a PSI-BLAST search within PyMod 2).

5 Usage Examples

Here we will provide a series of brief tutorials which will illustrate how to use PyMod 2 to perform some of the tasks the plugin was mainly designed to be used for.

5.1 Building a Single Chain Homology Model: Modeling of the Dihydrofolate Reductase from *Mycobacterium avium*

This tutorial will illustrate how to use PyMod 2 to build an homology model of the structure of a monomeric target protein, that is the dihydrofolate reductase of *Mycobacterium avium* (UniProt id: O30463), starting from its amino acid sequence.

5.1.1 Launch PyMod 2 and import the target sequence

5.1.1.A Retrieve the target sequence

- Use your web browser to reach the UniProt entry page of the target sequence: <http://www.uniprot.org/uniprot/O30463>.
- Download the sequence in the FASTA format: find the *Sequence* sections in the page, right-click on the *FASTA* button and use the *Save Link As* option of the pop-up menu of your browser. Save the sequence contained in these FASTA file on some location on your disk.

5.1.1.B Begin an new PyMod project

- Launch PyMOL and then start a new PyMod project by using the *Plugin* → *PyMod 2.X* option of PyMOL main window.

5.1.1.C Import the target sequence in PyMod

- Use the *File* → *Sequences* → *Open from File* option of PyMod main window and the FASTA file you have downloaded before. The sequence will be imported in PyMod as shown in **Figure 5.1**.

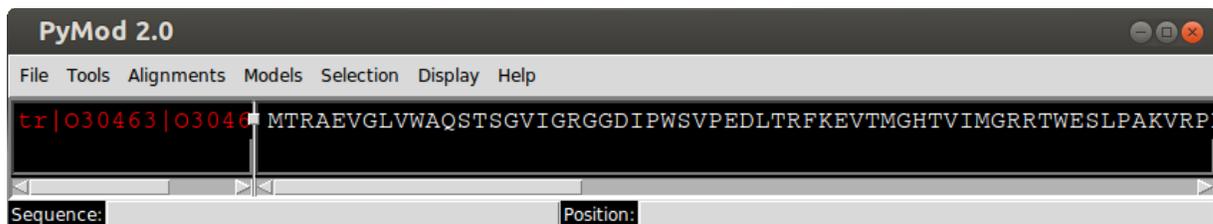


Figure 5.1. PyMod 2 main window with the target sequence loaded from a FASTA file.

5.1.2 Template searching and selection

- The first step of the homology modeling process is template identification of experimentally determined structures of proteins homologous to our target sequence, that is, template identification. To identify some templates we will use the BLAST algorithm.

5.1.2.A Use BLAST to search for templates

- To start a BLAST search within PyMod, select the target sequence by left-clicking on its header on PyMod main window left pane (the header will change its color from red to green once selected) and then use the **Tools** → **Database Search** → **BLAST** option on PyMod main menu.
- A BLAST preference window will appear (see **Figure 5.2**). It is possible to modify several parameters, however, in this tutorial we will keep the default values and just start the search by pressing the **Submit** button of the window. We can just point out that, as we are searching for templates 3D structures, the database parameter has to be set to *PDB*.

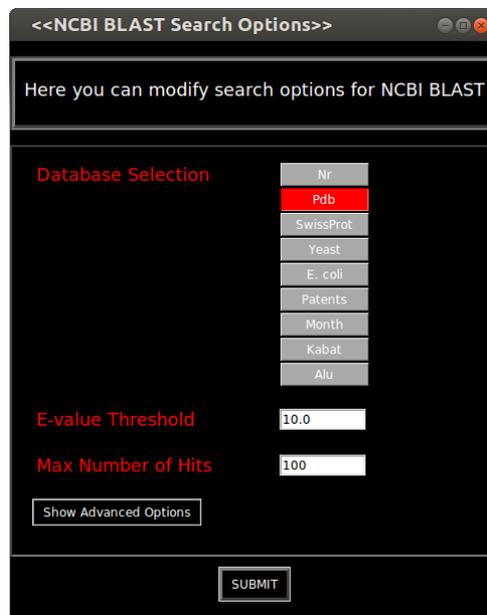


Figure 5.2. BLAST preferences window.

- PyMod will contact the NCBI server to perform the search. The operation could take some minutes. After the search has been completed, a results window will be displayed (see **Figure 5.3**).

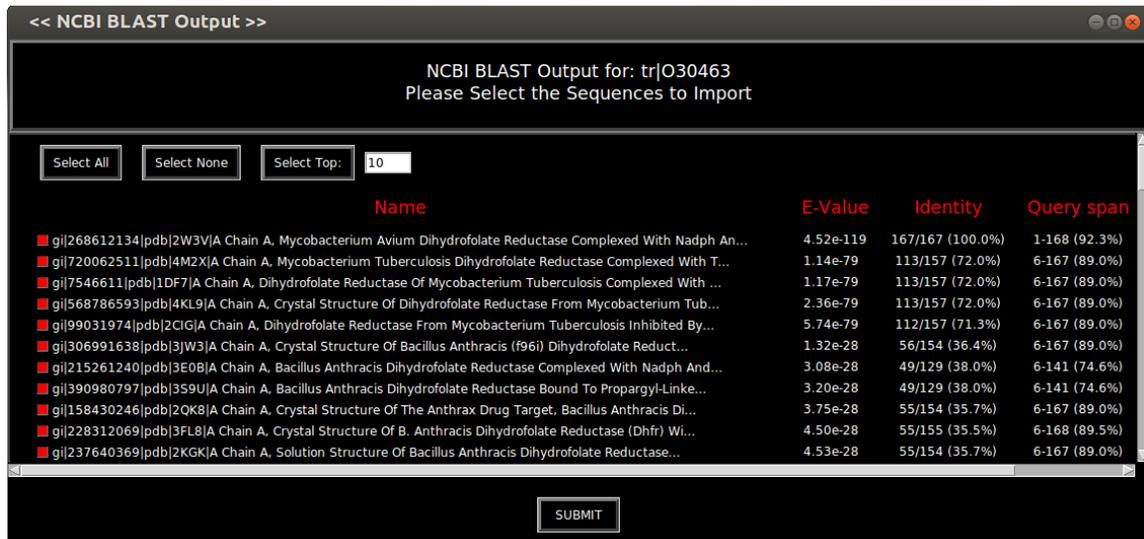


Figure 5.3. BLAST output window.

5.1.2.B Template selection

- As you can see from the results, the first hit sequence has 100% sequence identity with our query sequence. This is due to the fact that the structure of DHFR from *Mycobacterium avium* has already been experimentally determined. We will ignore this entry and use it later to validate our results. In this tutorial we will use two templates to build our model: DHFR from *Bacillus anthracis* (PDB id: 3JW3) and DHFR from *Moritella profunda* (PDB id: 2ZZA). Select these templates by clicking on their names and import their sequences in PyMod by pressing the results windows **Submit** button (you may have to scroll down a bit in the results list before you find those two templates).
- The sequences will be imported in PyMod main window, and clustered with your query sequence. You can expand or collapse this cluster by clicking on the + button placed beside your query sequence.

5.1.2.C Import the templates 3D structures

- To import the 3D structures of the templates, expand your cluster and right-click on each template sequence header and select the **Structure** → **Fetch PDB file** option.
- A dialogue will appear asking you to select the import mode (see **Figure 5.4**). Since we want to build our model of DHFR including its cofactor (an NADP⁺ molecule), we will choose the **Import all chains** option, because this import mode will load all the content of the template structure files, including their heteroatomic residues (such as their cofactors).

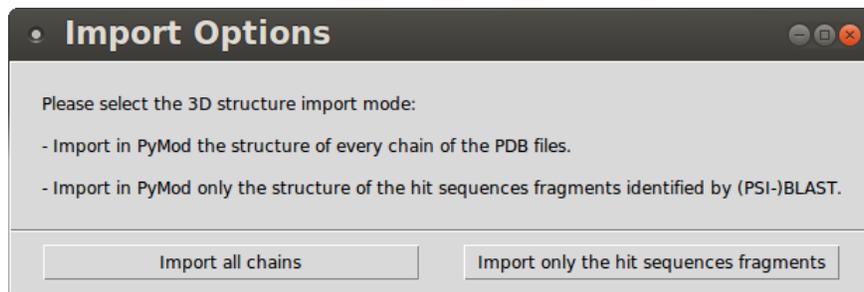


Figure 5.4. PDB structures import mode dialog.

- After a few seconds PyMod will import the structures and load them in PyMOL.
- By using the *Import all chains* import mode, the hit sequences identified by BLAST will be extracted from the query's cluster and replaced with the sequences of the template chains of the PDB files we fetched. The 3D structures of these chains will also be loaded in PyMOL (see **Figure 5.5**).

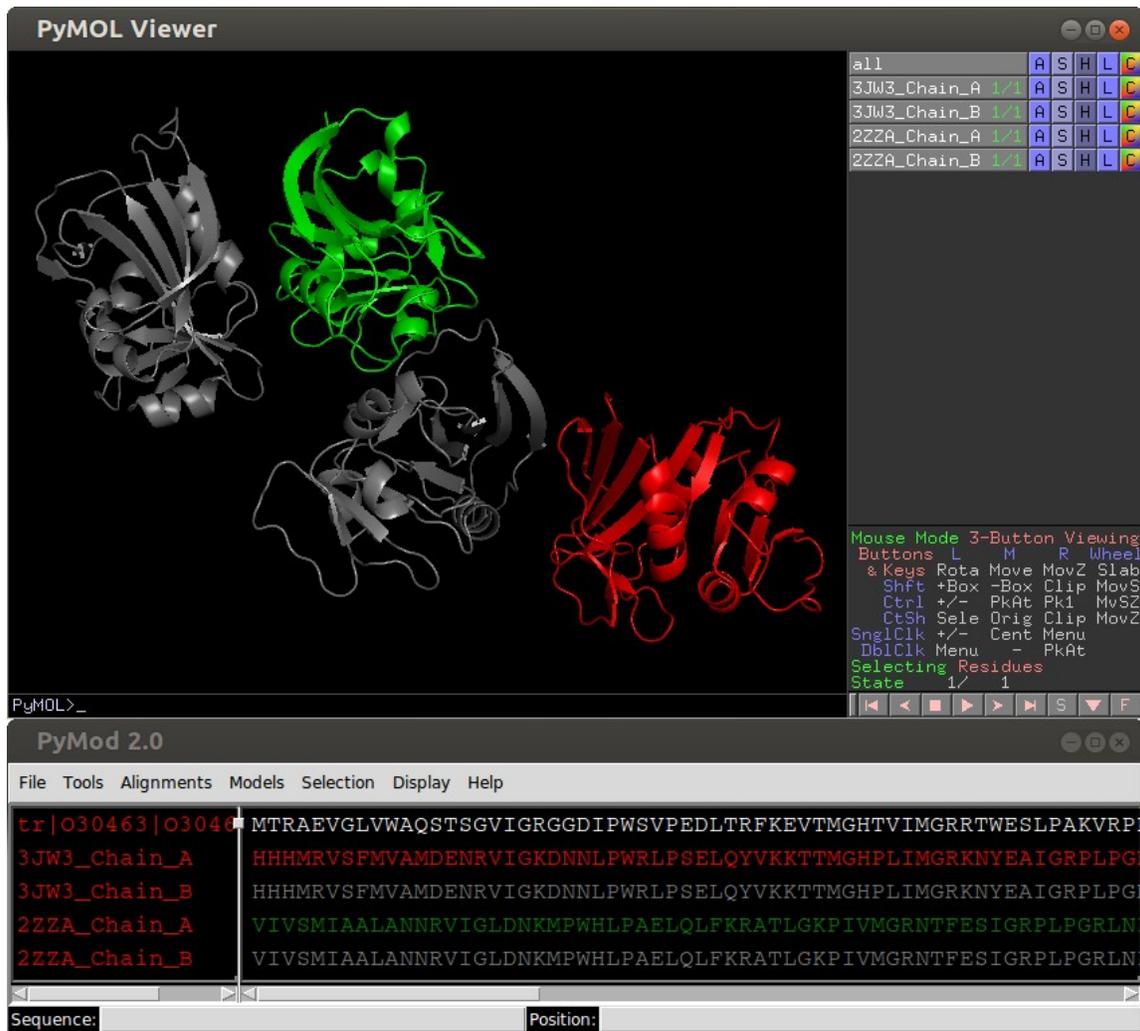


Figure 5.5. Templates structures imported in PyMOL and split by chain.

- Notice that in this case, for each hit sequence, two template chains are imported. This is because each of the template PDB files contain two almost identical chains of DHFR. These are not “actual dimers,” it is just that the asymmetric unit of the crystallographic structures each contain two molecules of DHFR, but in solution the DHFR is actually a monomer (the biological units of 3JW3 and 2ZZA reported in the RCSB PDB are defined as monomers, see <http://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/biological-assemblies> for an explanation of the difference between asymmetric units and biological units).
- As we will need only one chain from each template PDB structure, you can select all the the B chains of 3JW3 and 2ZZA (the ones colored in gray) and delete them using the *Sequence* → *Delete Sequence* option in the pop-up menu of PyMod main window left pane. In this way we will leave only the A chains of the structures.

5.1.3 Alignment building and editing

- The next step of the homology modeling process is the building of the target-templates sequence alignment.

5.1.3.A Build a structural alignment between the template chains

- In this example we will use two template chains: *3JW3_Chain_A* and *2ZZA_Chain_A*. When modeling with multiple templates, it is sometimes useful to superpose them as a first step and then to derive a structure based sequence alignment.
- To accomplish this task, select the headers of *3JW3_Chain_A* and *2ZZA_Chain_A* and use the **Tools** → **Structural alignment** → **CE Alignment** on PyMod main menu. The structures will be superposed in PyMOL and the derived structure based sequence alignment will be shown in PyMod main window (as seen in **Figure 5.6**).

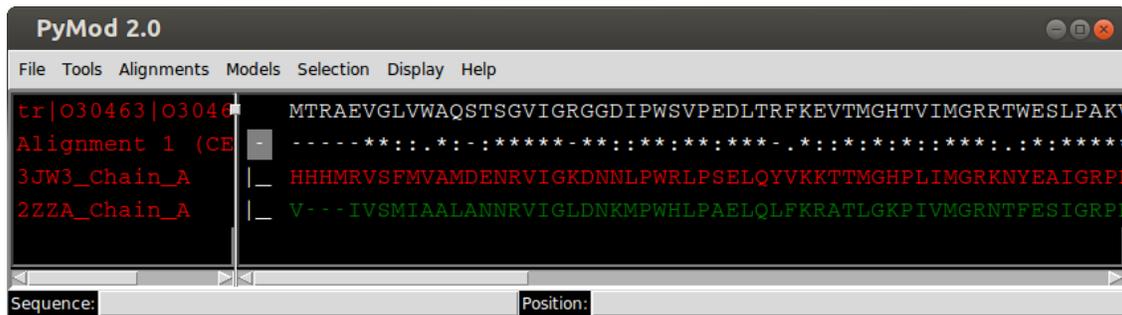


Figure 5.6. Structural alignment performed with the Combinatorial Extension algorithm.

5.1.3.B Align the target sequence to the structurally derived profile

- After the templates have been aligned, we will add the query sequence to the alignment using a sequence to profile alignment (our previously derived alignment represents in this tutorial a minimal structure-derived sequence profile). To accomplish this task you can choose between three different tools: ClustalW, MUSCLE or Clustal Omega. In this tutorial we will use Clustal Omega. Select all the sequences by left-clicking on the header and use the **Tools** → **Profile alignment** → **Clustal Omega** option. As usual, a preferences window will appear allowing you to modify of the parameters algorithm. We can just keep values at their default and perform the alignment by pressing the **Submit** button of the window. At this point the three sequences will be merged in the same cluster.

5.1.3.C Remove the C-terminal overhang from the target sequence

- Before we proceed to the model building phase, we notice that the last 13 amino acids of our target sequence are not aligned to any region of the templates. As terminal overhangs like this can not usually be accurately modeled by MODELLER, we will delete this region. Right-click

on the target's sequence header and use the *Sequence* → *Edit Sequence* option. In the sequence editing window select the last 13 amino acids and delete them with your keyboard (as shown in **Figure 5.7**). By pressing the *Submit* button of the window the sequence will then be updated.

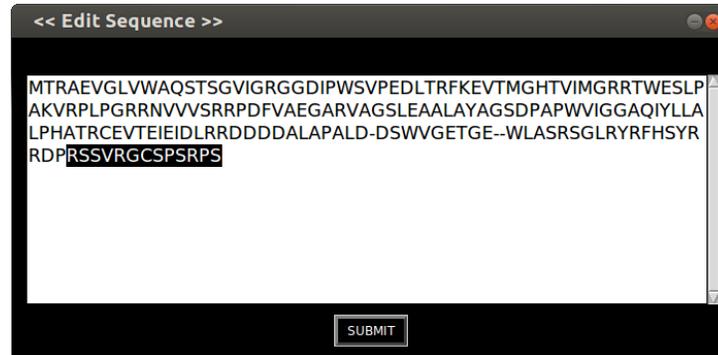


Figure 5.7. Sequence editing window. Make sure to delete the stretch of residues at the C-terminus highlighted in the figure.

5.1.4 Model building

- After target-templates alignment has been obtained, we can proceed with the last step of the homology modeling process, model building.

5.1.4.A PyMod homology modeling window

- In order to use MODELLER to build our homology models, select the query sequence and access the *Tools* → *Homology modeling* → *MODELLER* option of PyMod main menu. This will display the PyMod modeling window (see **Figure 5.8**).



Figure 5.8. Modeling options window.

- You will find several tabs in this window: in each one of them you might modify a series of parameters of the model building phase. In the *Main* tab of the window we can select our templates.

5.1.4.B Set the templates options for 2ZZA_Chain_A

- As a first step, select the *2ZZA_Chain_A* template by clicking on its the *Use as Template* check-button (see **Figure 5.8**). Its heteroatomic residues and water molecules options will be activated. By clicking on the *Select single heteroatomic residues* option we can see that *2ZZA_Chain_A* has three heteroresidues (see **Figure 5.9.a**).
 - NAP: this is the cofactor (NADP⁺) of the DHFR in an oxidized state. As we want to build our model with its cofactor, we will select this heteroresidue by clicking on its check-button.
 - FOL: this is a folate molecule bound in the active site of *2ZZA_Chain_A*. We are not interested in including it in our model, so we leave its check-button unchecked.
 - 1PE: this is just a surfactant molecule (pentaethylene glycol) used in the crystallization process and, since it has no biological meaning, we will not include it in our model by not

selecting its check-button.

- We would also like to include in our model this template model molecules, so we select the ***Include water*** check-button of this template.

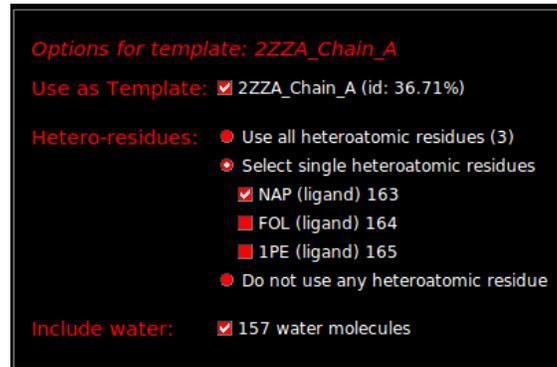


Figure 5.9. Template options for 2ZZA_Chain_A. Make sure that you set the options of these two templates as shown in the figure.

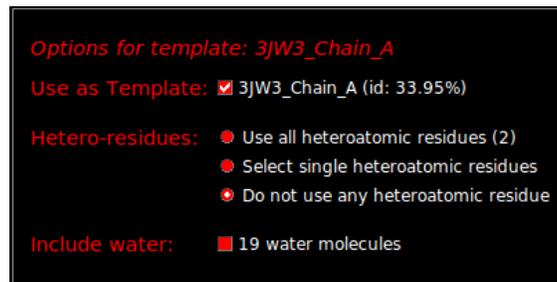


Figure 5.10. Template options for 3JW3_Chain_A.

5.1.4.C Set the templates options for 3JW3_Chain_A

- We also would like to use as a template 3JW3_Chain_A, so we click on its check-button. When we click on its ***Select single heteroatomic residues*** option, we see that it contains two hetero residues (see Figure 5.10).
 - NDP: the cofactor (NADPH) of DHFR in the reduced state. As we have already selected the NADP⁺ molecule from the 2ZZA_Chain_A template, we decide to leave this one unchecked. By selecting it, MODELLER would include it in the model along with the previously selected NADP⁺. As the two molecules occupy the same binding site of DHFR, including them both will lead to unrealistic model with two superposed ligands. We will thus leave it out by not selecting its check-button.

- TOP: that is, trimethoprim, an antibiotic. We are not interested in including it in our model, so we leave its check-button unchecked.
- Since we have decided not to use any heteroatomic residue of *3JW3_Chain_A*, we just select its ***Do not use any heteroatomic residue*** option of this template. We will also leave unchecked its ***Include water*** option.

5.1.4.D Set other modeling options

- In this tutorial we are interested in disulfide bridges building (DHFR is a cytosolic protein and does not contain any), so we skip ***Disulfides*** tab of the modelling window.
- As a last step before model building we set some important options on the ***Option*** tab of the modeling window. First we decide to set to *High* the ***Optimization Level*** option. This will lead to an higher degree of energy minimization of our model, and might increase accuracy in some of its regions. We also decide to set to *DOPE Score* the ***Color models by*** option. This will color the models according their residues DOPE (**D**iscrete **O**ptimized **P**rotein **E**nergy) value computed by MODELLER.
- After having set these options, click the ***Submit*** button on the modeling window to launch MODELLER. The modeling building phase could take several minutes.

5.1.5 Model analysis

- Once MODELLER has completed the model building phase calculations, the results will be displayed in PyMod/PyMOL.

5.1.5.A Models assessment table

- A table reporting the objective function value and the total DOPE value of each model will be displayed. These values are only useful in comparative terms, that is, when we build multiple copies of the same model and we want to select the one which appears to be the most accurate. Since we have built only one copy of our model we can ignore these values and close this window.

5.1.5.B DOPE profiles

- DOPE score based graph will appear for an energetic validation to the model (see **Figure 5.11**). As we can see from this graph, there seems to be some modeling mistakes (corresponding to high energy peaks of our model relative to the templates energy profiles). You might click on any region of the DOPE lines in order to highlight the corresponding residue in PyMOL.

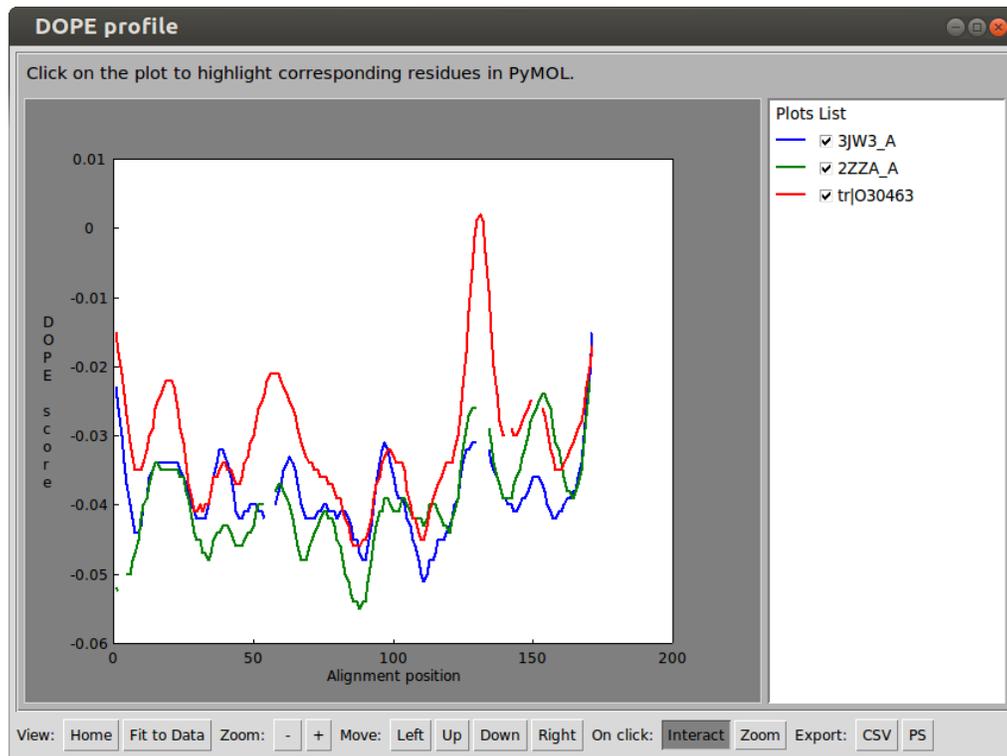


Figure 5.11. DOPE profile window.

5.1.5.C Inspecting the model 3D structure in PyMOL

- The 3D structure of the model will be imported in PyMOL. By inspecting it, we can see that the model core region and most part of the active site (where NADP⁺ is bound) are likely to be modeled correctly (they have low DOPE values on average and are colored in blue), while some loops appear to contain some modeling mistakes (they have high DOPE values on average and are colored in blue).

5.1.5.D Compare the model with the experimentally structure

- We can confirm this by confronting our model of DHFR with its experimentally determined structure. To load this structure in PyMOL we use the **Plugin** → **PDB Loader Service** command from PyMOL main windows and type 2W3W.
- The structure will be loaded in PyMOL. To import it into PyMod, use the **File** → **Sequences** → **Import PyMOL Objects** option from PyMod main window menu. On the preference window that will open, select the 2W3W object and press **Submit**.
- To superpose the experimentally determined structure to our model, select them in PyMod main window and use the **Tools** → **Structure alignment** → **Superose** function. The structures will be superposed in PyMOL as shown in **Figure 5.12**.

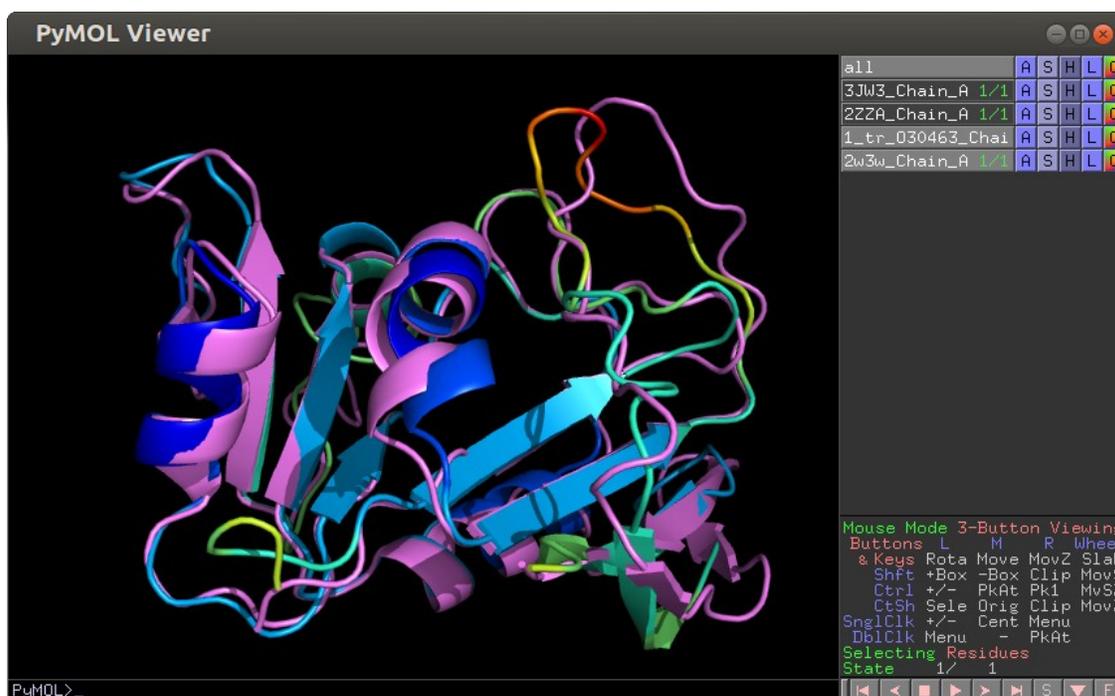


Figure 5.12. Superposition of the obtained model with the experimentally-solved 3D structure. In rainbow scale (with red corresponding to high DOPE values, blue to low DOPE values): model of the dihydrofolate reductase of *Mycobacterium avium*. In a violet: experimentally-solved structure of the same protein (PDB id: 2W3W).

- As we can see, our assumptions about the local quality of the model are confirmed. It is also important to stress that the possibility to build a model including the heteroatomic residues of the templates allows the right orientations of side chains in enzymes active sites, as shown in **Figure 5.13**.

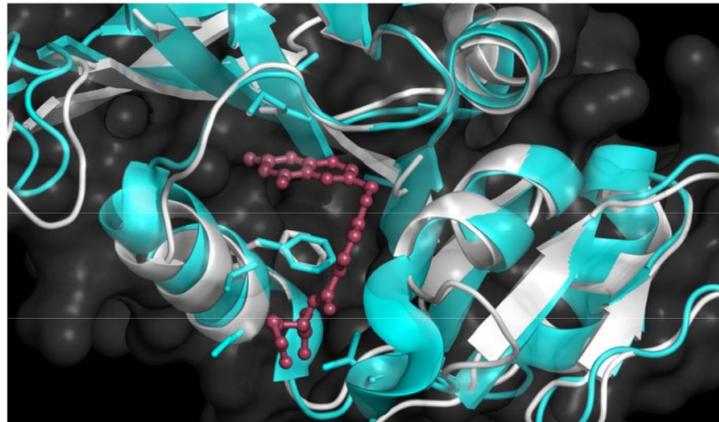


Figure 5.13. In the picture is shown the correct orientation of side chains that interact with the cofactor in the active site of the protein. In white: model of the dihydrofolate reductase of *Mycobacterium avium*. In cyan: experimentally-solved structure of the same protein (PDB code: 2W3W).

5.2 Building a Multiple Chain Homology Model: Modeling of Human Hemoglobin Portland I

This tutorial will illustrate how to use PyMod 2 to build an homology model of a protein complex, in this case, human hemoglobin Portland I (herein HbE-P1). HbE-P1 is a form of the oxygen carrying hemoglobin existing at low levels in the blood during embryonic and fetal life. It is a tetramer composed of two ζ and γ hemoglobin fold chains. Currently no experimentally determined structure of HbE-P1 exists and we will try to build an homology model of it.

Homology modeling of protein complexes (such as HbE-P1) holds its basis in the fact that, for high levels of sequence similarity, homologous protein complexes retain similar architectures [Aloy et al., 2003]. Thus, homology modeling is likely to build accurate models when we have high levels of sequence conservation between homologous members of target and template complexes. For cases of low sequence similarity, architectures start to lose their conservation and in order to predict the structure of target a complex other computational approaches (such as protein-protein docking) might be preferred. Here we will review the steps necessary to build a multiple chain homology model in PyMod 2.

5.2.1 Launch PyMod and import the target sequences

5.2.1.A Retrieve the target sequences

- Use your web browser to reach the UniProt entry page of the target sequences:

- <http://www.uniprot.org/uniprot/P02008> for chain ζ .
- <http://www.uniprot.org/uniprot/P69891> for chain γ .
- Download the sequences in the FASTA format: find the *Sequence* sections in both pages, right-click on the *FASTA* buttons and use the *Save Link As* option of the pop-up menu of your browser. Save the sequences contained in these FASTA format files on some location on your disk.

5.2.1.B Begin an new PyMod project

- Launch PyMOL and then start a new PyMod project by using the *Plugin* \rightarrow *PyMod 2.X* option of PyMOL main window.

5.2.1.C Import the target sequences in PyMod

- Use the *File* \rightarrow *Sequences* \rightarrow *Open from File* option of PyMod main window and open both FASTA files you have downloaded before. The sequences will be imported in PyMod as shown in **Figure 5.14**.

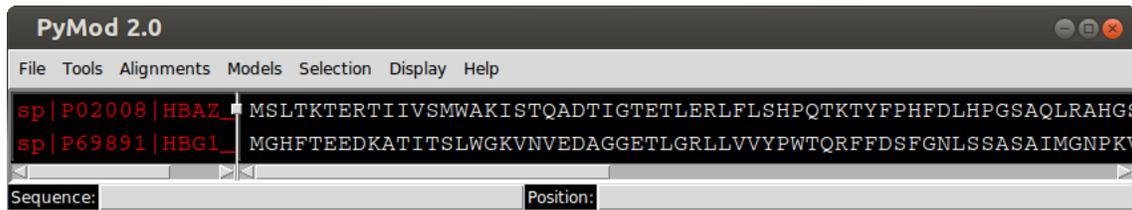


Figure 5.14. PyMod 2 main window with the target sequences loaded from FASTA files.

5.2.1.D Duplicate the target sequences

- Since HbE-P1 is a tetramer with an $\zeta_2\gamma_2$ stoichiometry, we will need two copies of each target sequence, so we will have to copy each one of them. To copy a sequence, right-click on its name on PyMod main windows left pane and use the *Sequence* \rightarrow *Duplicate sequence* option. By copying each target sequence we will obtain two copies of ζ chain and two copies of γ chain (as illustrated in **Figure 5.15**).

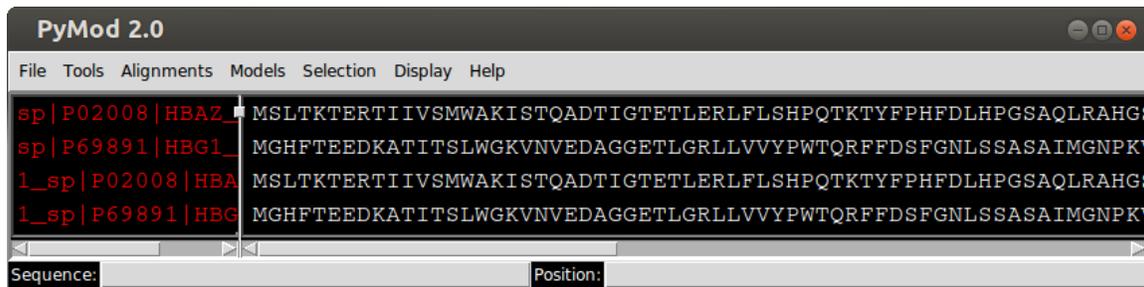


Figure 5.15. PyMod 2 main window with the two target sequences which have been duplicated.

5.2.2 Import the template complex in PyMod

- PyMod multiple chain homology modeling mode needs a “template complex”. A template complex can be any structure file containing multiple polypeptide chains. The template complex architecture (that is, the relative orientations and positions of its chains) will be used to build the homology model of target complexes.

5.2.2.A Retrieve the template complex

- The first step of homology modeling is template searching. However, in this tutorial, in order to make things quicker, we will skip this first step and just retrieve from the PDB our template complex file.
- We decide to use the structure with PDB id: 2DN2 as our template complex (if you perform a template search by yourself, you may find that this template is not the best possible one to model HbE-P1, but here it will be useful to illustrate how PyMod multiple chain homology modeling mode works). This crystallographic structure contains in its asymmetric unit a full hemoglobin A tetramer. Hemoglobin A is composed of two α and β hemoglobin-fold chains homologous to our ζ and γ chains, with the α chains being more closely related to the ζ chains and the β chains being more closely related to the γ chains. Thus, the two complex are homologous.
- To download the template complex structure file, go to the RCSB PDB entry page of the structure (<http://www.rcsb.org/pdb/explore/explore.do?structureId=2dn2>) and download its file in PDB format by using the **Download Files** \rightarrow **PDB format** option on the page.

5.2.2.B Import the template complex in PyMod/PyMOL

- Once you have downloaded the file, open it in PyMod by using the **File** \rightarrow **Sequences** \rightarrow **Open from File** command in the plugin main window. The four chains contained in this structure (two α chains named *2DN2_Chain_A* and *2DN2_Chain_B*, and two β chains named *2DN2_Chain_C* and *2DN2_Chain_D*) will be loaded in PyMod/PyMOL alongside our four target sequences (as

shown in **Figure 5.16**).

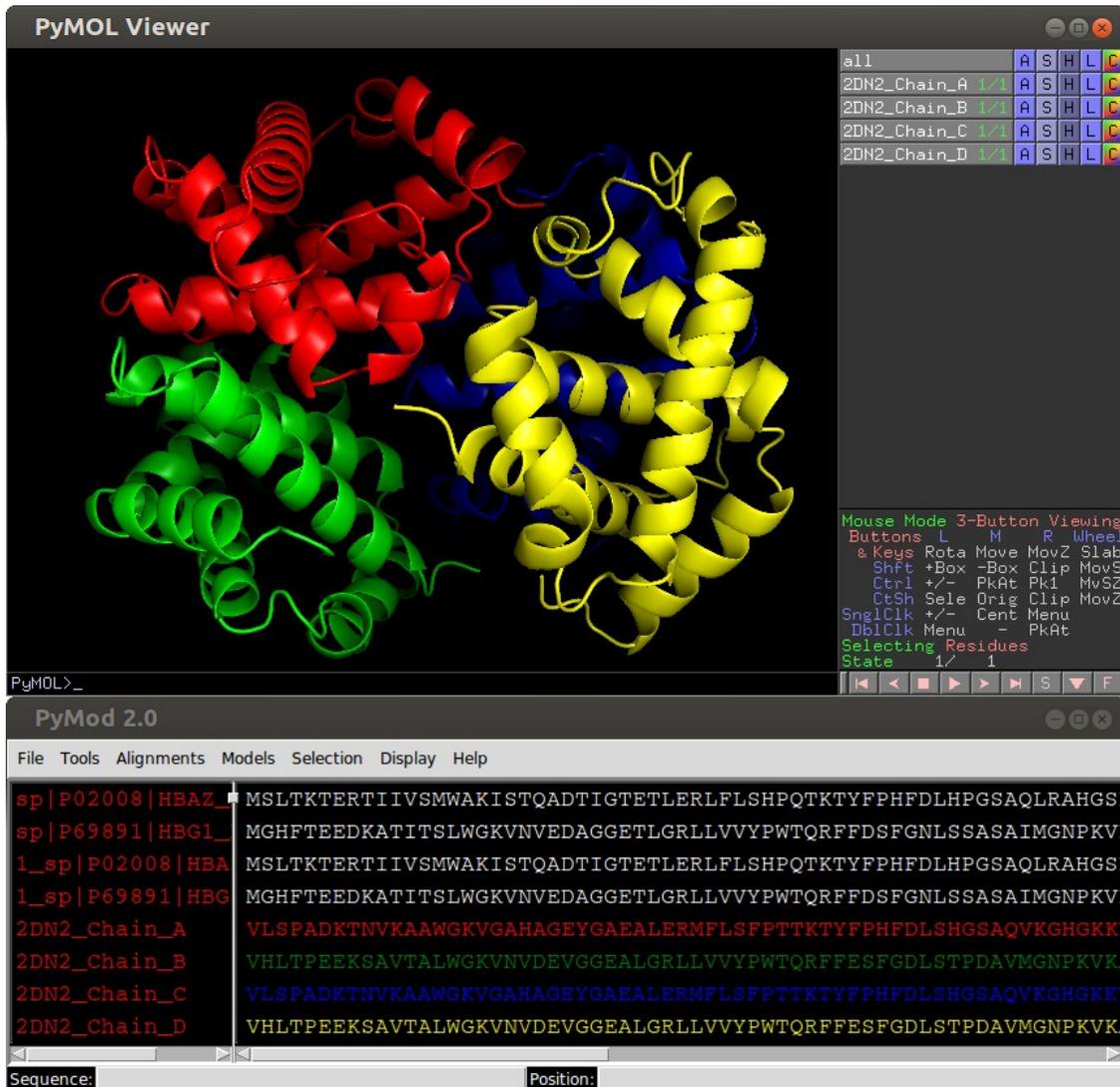


Figure 5.16. Templates structures imported in PyMOL and split by chain.

- Notice that this experimentally determined structure contains hemoglobin A in its T state (it is not binding oxygen), so what we will be building is actually a model of hemoglobin Portland I state T.

5.2.3 Target-template alignments building

- The second step in homology modeling is alignment building between the target and template sequences. In order to build a multiple chain homology model in PyMod, we have to align all the target sequences to the corresponding chains of the template complex.

- In this case we have to align our hemoglobin ζ chains to the α chains of the template complex and our γ chains to the corresponding β chains. Since there is a high level of sequence similarity between our target and template sequences (59.6% of sequence identity between ζ and α chains and 72.6% between γ and β chains), performing pairwise alignments between our sequences will most probably result in accurate alignments. For this reason we will follow this alignment strategy in this tutorial.
- Select sequence *sp|P02008|HBAZ_HUMAN* and *2DN2_Chain_A* (by clicking on their names in PyMod main windows left pane) and use the **Tools** → **Sequence Alignment** → **ClustalW** command. A preferences window will appear allowing you to modify of the parameters algorithm. We can just keep values at their default and perform the alignment by pressing the **Submit** button of the window. A pairwise sequence alignment will be performed and a cluster containing the two aligned sequences will be generated.
- Repeat the same operation for the other three target sequences. That is, align
 - *1_sp|P02008|HBAZ_HUMAN* to *2DN2_Chain_C*
 - *sp|P69891|HBG1_HUMAN* to *2DN2_Chain_B*
 - *1_sp|P69891|HBG1_HUMAN* to *2DN2_Chain_D*.
- Once you have built the four alignments, you should have four clusters as shown in **Figure 5.17**.

```

PyMod 2.0
File Tools Alignments Models Selection Display Help

Alignment 1 (Cl
sp|P02008|HBAZ_
2DN2_Chain_A
Alignment 2 (Cl
sp|P69891|HBG1_
2DN2_Chain_B
Alignment 3 (Cl
1_sp|P02008|HBA
2DN2_Chain_C
Alignment 4 (Cl
1_sp|P69891|HBG
2DN2_Chain_D

Sequence: sp|P69891|HBG1_HUMAN Hemoglobin subunit gamma Position: TRP 38 - Alignment: 38

```

Figure 5.17. PyMod 2 main window with the four target sequences, each aligned to the four templates.

5.2.4 Model building

- To build a multiple chain homology model in PyMod, we need to select at the same time all our target sequences (as shown in **Figure 5.18**) and then use the *Tools* → *Homology modeling* → *MODELLER* option from PyMod main menu.

```

PyMod 2.0
File Tools Alignments Models Selection Display Help

Alignment 1 (C1)  - -*:--::*--:-*.*:::*--*:***:***.*-*****--***:..
sp|P02008|HBAZ_2DN2_Chain_A | MSLTKTERTIIVSMWAKISTQADTIGTETLERLFLSHPQTKTYFPHFDLHPGSAQLR
- VLSPADKTNVKAAWGKVG AHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKK

Alignment 2 (C1)  - -*:--*:***:***.*-*****--***:..
sp|P69891|HBG1_2DN2_Chain_B | MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSASAIMGM
- VHLTPEEKSAVTALWGKVNVEVGG EALGRLLVVYPWTQRFFESFGDLSTPDAVMGM

Alignment 3 (C1)  - -*:--*:***:***.*-*****--***:..
1_sp|P02008|HBAZ_2DN2_Chain_C | MSLTKTERTIIVSMWAKISTQADTIGTETLERLFLSHPQTKTYFPHFDLHPGSAQLR
- VLSPADKTNVKAAWGKVG AHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKK

Alignment 4 (C1)  - -*:--*:***:***.*-*****--***:..
1_sp|P69891|HBG1_2DN2_Chain_D | MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSASAIMGM
- VHLTPEEKSAVTALWGKVNVEVGG EALGRLLVVYPWTQRFFESFGDLSTPDAVMGM

Sequence: sp|P02008|HBAZ_HUMAN Hemoglobin subunit zeta | Position: ARG 57 - Alignment: 57

```

Figure 5.18. In order to build a multiple chain homology model, you have to select at the same time all the target chains, as shown in the figure.

- The modeling window will open. To build our model we first need to select our template complex chains. For each target sequence, click on the *Use as Template* check-buttons of their templates chains (as shown in **Figure 5.19**).

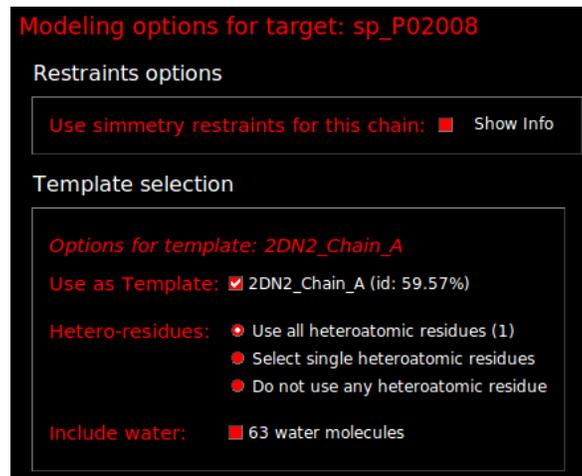


Figure 5.19. In order to build the model, you have to select for each target chain the corresponding template complex chain by clicking on their *Use as Template* checkbuttons.

- In this tutorial we are not interested in modifying other options, so we will leave them with the default values. We just point out that, for each template chain, we are leaving the *Use all heteroatomic residues* option selected: this will include in our model all the heteroatomic residues of the template chains (in this case, the heme molecules co-crystallized with the hemoglobin template complex). Let us proceed to build our model by clicking on the modeling window *Submit* button.
- This will launch MODELLER: homology modeling of multiple chain structures can take MODELLER a while (depending on the number of chains, their lengths and additional modeling parameters) so you might have to wait some minutes before MODELER finishes its calculations.

5.2.5 Model analysis

- Once MODELER has built our model, the results will be imported in PyMod/PyMOL.

5.2.5.A Models assessment table

- We saw the information provided by the assessment table in the first Usage Example of this guide (chapter [5.1](#)). In this case we will just point out that both the objective function and DOPE values are computed for the entire multiple chain model complex.

5.2.5.B DOPE profiles of multiple chain models

- We saw the information provided by the DOPE (**D**iscrete **O**ptimized **P**rotein **E**nergy) profile in

the first Usage Example of this guide (chapter [5.1](#)). In the DOPE graph of multiple chain models, the DOPE profiles of the target chains, together with their template chains, are placed one after the other (see [Figure 5.20](#)).

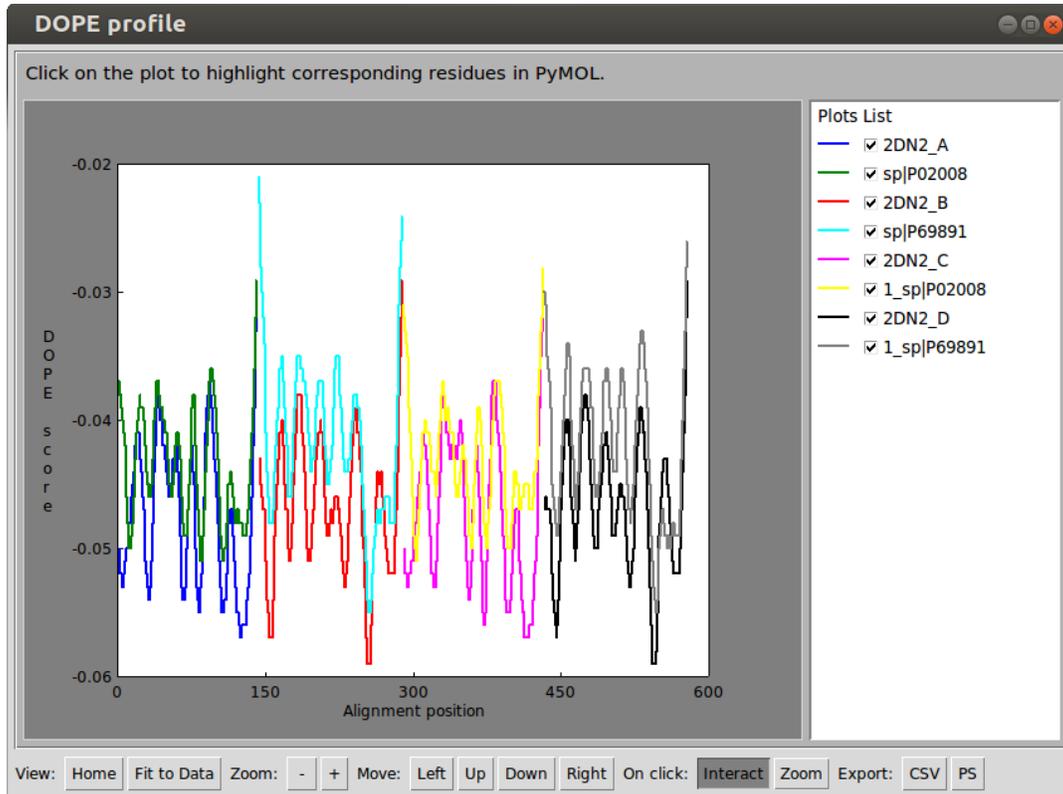


Figure 5.20. DOPE profile window.

Note that the DOPE function was designed to score the energy of single chain globular proteins. As the energy of inter-chain contacts can not be adequately scored by the DOPE function, DOPE profiles of multiple chain models are not useful for assessing the quality of whole model complexes. However, they remain useful for assessing the quality of the single chains of the model. Keeping in mind this, we see that the chains of our model have very similar profiles to those of the corresponding template chains. Considering that the quaternary structure of hemoglobin Portland is likely to be conserved with the structure of hemoglobin A (as there is an high level of sequence similarity among the corresponding members of the homologous complexes), this probably means that our model is a reasonable one.

5.2.5.C Inspecting the model complex in PyMOL

- By inspecting the structure of our model in PyMOL, we see that each model chain has its heme ligand bound in its heme-binding pocket (see [Figure 5.21](#)).

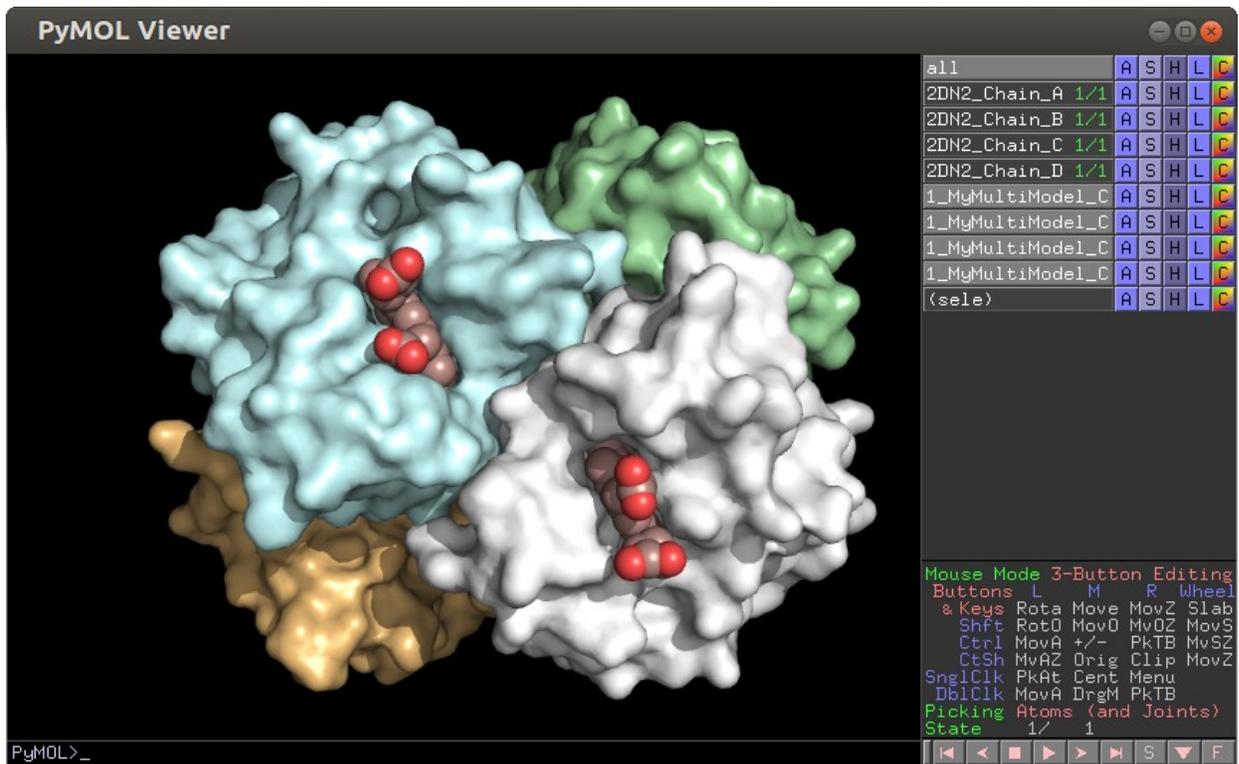


Figure 5.21. The heme molecules (colored in dark salmon) are included in the final model of the tetramer (shown as surface).

- This is due to the fact that, while we were setting the options of the modeling window, we let the *Use all heteroatomic residues* options in an active state: this informed MODELLER to include in our model the heme ligands of the template. It is important to stress that the possibility to easily build homology models of complex systems (multiprotein complexes containing heteroatomic moieties) can facilitate the building biological meaningful models.

6 PyMod 2 Development

6.1 General Information and Contacts

PyMod 2 is distributed under the LGPL. The source code of the PyMod project is hosted on GitHub at: <https://github.com/giacomojanson/pymod>.

The PyMod project is actively under development. For updates on this project, please visit regularly the home page of the project at: <http://schubert.bio.uniroma1.it/pymod/index.html>.

If you wish report bugs or collaborate to PyMod development with new ideas and/or code, you are more than welcome. For any kind of question you can contact us at: giacomo.janson@uniroma1.it.

6.2 List of Tested Platforms

We have tested thoroughly PyMod 2 and the **PyMod Installer Bundle** on as many PyMOL builds as possible. In the **Table 6.1** you will find a list of the PyMOL builds on which we have successfully tested PyMod 2 (along its Installer Bundle). We have made every effort in order to develop PyMod 2 as portable as possible and the plugin will certainly run on many other PyMOL builds. However, since there is a huge number of PyMOL builds around, there is a chance that your PyMOL build will not support the plugin. If PyMod 2 is not running on your PyMOL, please let us know by sending information (see the section above for contacts) about your system and PyMOL build (you can provide us the type of information reported in the table below), so that we may add compatibility to that particular PyMOL build.

Operating system	OS arch.	PyMOL version	PyMOL arch.	PyMOL build type	Python version	Has NumPy	Has Biopython	PyMod 2 support	Installer Bundle support	Notes
Windows 8	64	1,81	64	Incentive	2,7	Yes	No	Yes	Yes	
Mac OS X	64	1,82	64	Incentive	2,7	Yes	Yes	Yes	Yes	
Linux, Ubuntu 14.04	64	1,72	64	Unofficial (apt-get)	2,7	apt-get	apt-get	Yes	Yes	

Table 6.1. Systems on which we tested PyMod 2 and its Installer Bundle. The *Has NumPy* and *Has Biopython* columns indicate if the NumPy and Biopython libraries are included in the PyMOL build. If the libraries are easily installable through a package manager and can be imported by the PyMOL build, the name of the package manager is reported (for example *apt-get* for Debian based distributions).

7 Appendix

7.1 Locate PyMOL Default Plugin Directory

In order to know the path of PyMOL plugin directory, you may use the following instructions. Open PyMOL and type in the PyMOL console the following command:

```
import os; import pmg_tk; print os.path.dirname(pmg_tk.__file__)
```

You will see a path displayed in the output region of the PyMOL main window, specifying the *pmg_tk* directory of PyMOL. Inside this directory you will find a *startup* directory. This *startup* directory is the default plugin directory of your PyMOL build.

7.2 Manually Install PyMod 2

In the **Installation Instructions** section we described how to install PyMod 2 on your system using the **PyMod Installer Bundle**. However, you may also install it manually (as any other PyMOL plugin) by following the instructions below. This might be useful if you intend to have fine control over its installation location, for example if you wish to develop new PyMod 2 code.

7.2.1 For PyMOL versions with a Plugin Manager (PyMOL >= 1.5.0.5)

- Download the PyMod 2 plugin zip file at <http://schubert.bio.uniroma1.it/pymod/download.html>.
- Since version 1.5.0.5, PyMOL has an easy-to-use Plugin Manager, which you can use to install PyMod 2 (you can verify the presence of the Plugin Manager on your PyMOL built by checking if the *Plugin* → *Plugin Manager* command exists in the main menu of PyMOL main window).
- Launch the PyMOL Plugin Manager through the *Plugin* → *Plugin Manager* command in PyMOL main window. Go to the *Install New Plugin* tab of the Plugin Manager window and use the *Choose file...* option (as seen in **Figure 7.1**). Finally, select the PyMod .zip file you downloaded previously to install the plugin.

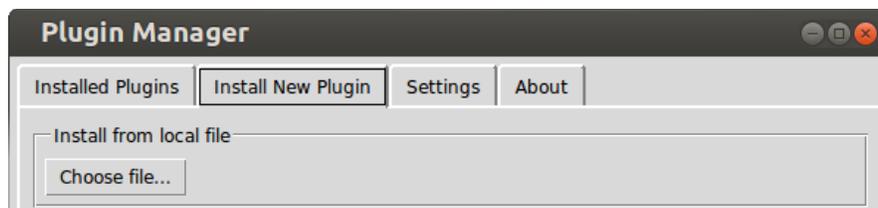


Figure 71. Part of the PyMOL Plugin Manager window. Press the *Choose file...* button and select the PyMod 2 zip file in order to install it on PyMOL.

- Make sure to quit PyMOL before you proceed to the next installation step.

7.2.1.A For PyMOL versions without a Plugin Manager (PyMOL <= 1.5.0.5)

- Older PyMOL versions do not have a Plugin Manager. In these versions, you will not be able to install PyMod 2 through the **Plugin** → **Manage Plugins** → **Instal...** (or in some versions **Plugin** → **Install Plugin...**) command of PyMOL main menu, because this command only installs single file plugins and PyMod 2 is a multiple file plugin. In order to install PyMod 2 on older PyMOL versions you download the PyMod 2 plugin zip file at <http://schubert.bio.uniroma1.it/pymod/download.html> and unzip its contents (a folder named *pymod2* and a file named *pymod2.py*) in the PyMOL plugin directory (see **Locate PyMOL Default Plugin Directory**).

7.3 PyMod Installer Bundle

PyMod 2 depends on the NumPy and Biopython libraries. Additionally its full functionalities can be accessed only if a set of external bioinformatics tools (see **Table 1.1**) is installed on your system. In order to facilitate the installation of these components, we provide a **PyMod Installer Bundle**.

7.3.1 PyMod Installer Bundle Usage

7.3.1.A Retrieve the PyMod Installer Bundle

- To use the PyMod Installer Bundle, download the latest version of its zipped file at <http://schubert.bio.uniroma1.it/pymod/download.html>. Choose the right version for your operating system (Windows, Mac OS X or Linux).
- Once you have downloaded it, unzip its content in any folder of your system. In the unzipped directory you will find some Python scripts (files ending in .py) and a zipped file containing all the installation files. You can use the Python scripts in order to install and configure PyMod 2 and its components on your system.

7.3.1.B Launch Installer Bundle Scripts

- To run any of these Python scripts, simply launch PyMOL and use the **File** → **Run...** command present in the main menu of PyMOL main window (see **Figure 7.2**).

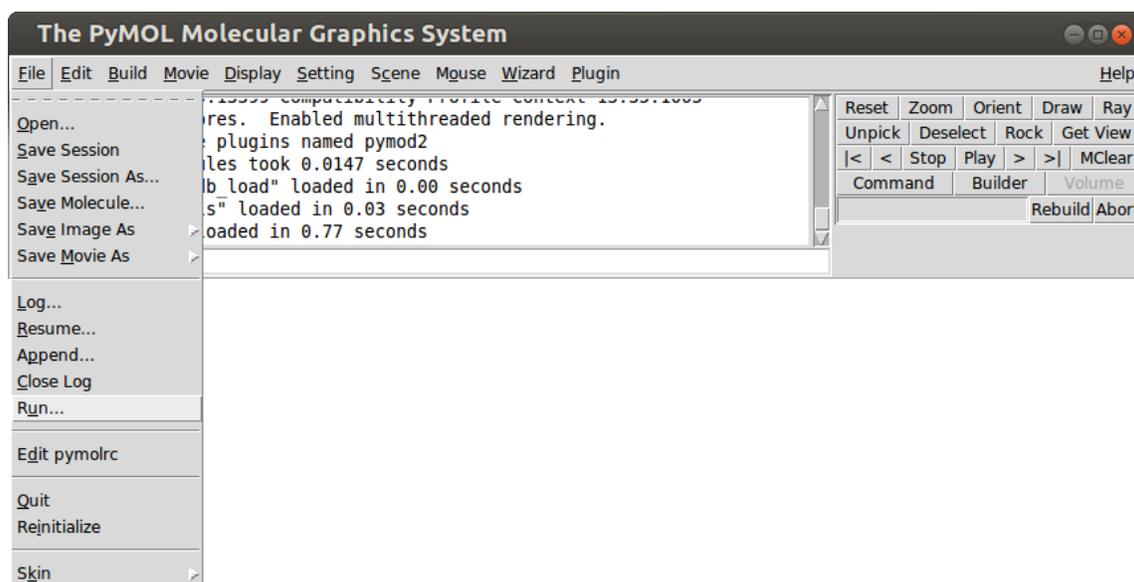


Figure 7.2. Select the **Run...** command in the **File** submenu of PyMOL main window to run the **PyMod Installer Bundle** scripts.

- A file dialog will appear: by selecting one of the scripts of the **PyMod Installer Bundle** (see **Figure 7.3**), you will be able to execute it. In the following sections, we will explain how the scripts available in the Installer Bundle can be used.

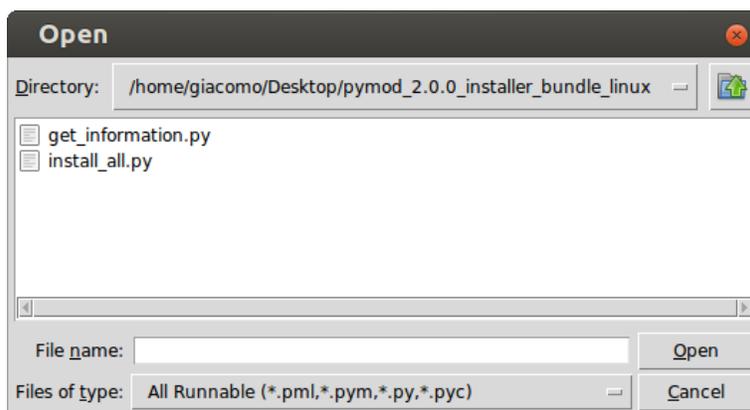


Figure 7.3. Select on the file picker one of the Python scripts in the **PyMod Installer Bundle** folder in order to run it.

7.3.2 install_all.py

7.3.2.A General information

- By running this script you will install on your system the following components:

- PyMod 2.
- NumPy and Biopython libraries.
- ClustalW, Clustal Omega, MUSCLE, PSI-BLAST, PSIPRED and ksdssp executable and data files.
- SwissProt and PDB sequence databases (derived from <ftp://ftp.ncbi.nlm.nih.gov/blast/db/>).
- The **install_all.py** script will check on your system for the existence of any of components listed above. The script will only install missing components. If any of them is already installed, the script will not reinstall them, but it will configure PyMod 2 in order to use their already existing versions. Additionally, once you have installed some components with the **Installer Bundle**, you may easily substitute them with your own version at a later moment (through the use of the **PyMod 2 Options** window). Indeed, PyMod 2 has a modular nature: you may install some of its external bioinformatics tools manually (for example through the packages provided by the authors of the programs, or through your package management system) and at the same time install some tools using the **PyMod Installer Bundle**.

7.3.2.B Usage

Please refer to the **Installation Step 2** parts of the **Installation Instructions** to learn how to launch this script on your OS.

7.3.2.C Note on install_all.py script usage

You can run this script at any moment after having manually installed the PyMod 2 plugin (see **Manually Install PyMod 2** to learn how). You may also run this script before installing PyMod 2. In this case the script will also take care of the installation of PyMod 2 files in PyMOL default plugin directory and then it will proceed to install the additional components. However, we recommend running this script after having manually installed PyMod 2, because by doing that, you will have control over PyMod 2 its target installation directory (this might come in handy if you want to remove or update PyMod 2 in the future).

7.3.2.D install_all.py target directories

In the following section you will find information about where the files of the components installed through the **install_all.py** script are going to be placed on your system.

- PyMod 2: its files will be placed in the default plugin directory of PyMOL which is usually located in some location of your file system under administrator privileges. If, while previously installing other PyMOL plugins, you have defined another plugin directory in some location of your filesystem which does not require administrator privileges to be written, the installer script will place PyMod 2 files in this directory. In this way removal or update of PyMod will result quicker.

- NumPy and Biopython: their files will be placed in the parent directory of the PyMOL *png_tk* directory, so that NumPy and Biopython versions installed through the **install_all.py** can only be used within PyMOL.
- ClustalW, Clustal Omega, MUSCLE, PSI-BLAST, PSIPRED and ksdssp: the executable and data files of these tools will be placed in the *external_tools* folder in your **PyMod Directory**. In this way,
- SwissProt and PDB sequence databases: their data files (447 megabytes) will be placed in a *data/blast_databases* folder in your **PyMod Directory**.

7.3.3 get_information.py

- This is an additional script shipped in the **PyMod Installer Bundle** which will not install anything on your system, but it will provide you with information about your PyMOL build (version, architecture, etc...) and the installation status of PyMod 2 and its components.
- To run the **get_information.py** script you will need to launch PyMOL and use the **File** → **Run...** command from PyMOL main menu and to select its file in the PyMod Installer Bundle directory (for details see **Launch Installer Bundle Scripts**).
- The script will print a series of information in the output region of the PyMOL main window. You can scroll the output using the window scrollbar. Among the useful information which you might find, there is the architecture of your PyMOL build (listed in the '*PyMOL information*' section of the output) and the installation status of the PyMod 2 components on your system (listed in the '*Checking components installation state*' section of the output).

8 Bibliography

- Aloy, P., Ceulemans, H., Stark, A., and Russell, R.B. (2003). *The relationship between sequence and interaction divergence in proteins*. *J. Mol. Biol.* 332, 989–998.
- Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. (1997). *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. *Nucleic Acids Res.* 25, 3389–3402.
- Blackshields, G., Sievers, F., Shi, W., Wilm, A., and Higgins, D.G. (2010). *Sequence embedding for fast construction of guide trees for multiple sequence alignment*. *Algorithms Mol Biol* 5, 21.
- Bramucci, E., Paiardini, A., Bossa, F., and Pascarella, S. (2012). *PyMod: sequence similarity searches, multiple sequence-structure alignments, and homology modeling within PyMOL*. *BMC Bioinformatics* 13 Suppl 4, S2.
- Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T.L. (2009). *BLAST+: architecture and applications*. *BMC Bioinformatics* 10, 421.
- Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. (2009). *Biopython: freely available Python tools for computational molecular biology and bioinformatics*. *Bioinformatics* 25, 1422–1423.
- Crooks, G.E., Hon, G., Chandonia, J.-M., and Brenner, S.E. (2004). *WebLogo: a sequence logo generator*. *Genome Res.* 14, 1188–1190.
- Edgar, R.C. (2004). *MUSCLE: multiple sequence alignment with high accuracy and high throughput*. *Nucleic Acids Res.* 32, 1792–1797.
- Janson, G., Zhang, C., Prado, M.G., and Paiardini, A. (2016). *PyMod 2.0: improvements in protein sequence-structure analysis and homology modeling within PyMOL*. *Bioinformatics*.
- Jones, D.T. (1999). *Protein secondary structure prediction based on position-specific scoring matrices*. *J. Mol. Biol.* 292, 195–202.
- Kabsch, W., and Sander, C. (1983). *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features*. *Biopolymers* 22, 2577–2637.
- Madhusudhan, M.S., Webb, B.M., Marti-Renom, M.A., Eswar, N., and Sali, A. (2009). *Alignment of multiple protein structures based on sequence and structure features*. *Protein Eng. Des. Sel.* 22, 569–574.
- Paiardini, A., Bossa, F., and Pascarella, S. (2005). *CAMPO, SCR_FIND and CHC_FIND: a suite of web tools for computational structural biology*. *Nucleic Acids Res.* 33, W50-55.
- Robert, X., and Gouet, P. (2014). *Deciphering key features in protein structures with the new ENDscript server*. *Nucleic Acids Res.* 42, W320-324.
- Sali, A., and Blundell, T.L. (1993). *Comparative protein modelling by satisfaction of spatial restraints*. *J. Mol. Biol.* 234, 779–815.
- Schrödinger, LLC. (2015, November). *The PyMOL Molecular Graphics System, Version 1.8*.
- Shen, M.-Y., and Sali, A. (2006). *Statistical potential for assessment and prediction of protein structures*. *Protein Sci.* 15, 2507–2524.
- Shindyalov, I.N., and Bourne, P.E. (1998). *Protein structure alignment by incremental combinatorial extension (CE) of the optimal path*. *Protein Eng.* 11, 739–747.

- Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., et al. (2011). *Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega*. *Mol. Syst. Biol.* 7, 539.
- Söding, J. (2005). *Protein homology detection by HMM-HMM comparison*. *Bioinformatics* 21, 951–960.
- Talevich, E., Invergo, B.M., Cock, P.J.A., and Chapman, B.A. (2012). *Bio.Phylo: a unified toolkit for processing, analyzing and visualizing phylogenetic trees in Biopython*. *BMC Bioinformatics* 13, 209.
- Thompson, J.D., Higgins, D.G., and Gibson, T.J. (1994). *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*. *Nucleic Acids Res.* 22, 4673–4680.
- Van der Walt, S., Colbert, S.C., and Varoquaux, G. (2011). *The NumPy array: a structure for efficient numerical computation*. *Computing in Science & Engineering* 13, 22–30.