MODELLER - I - Introduction

Jean-Yves Sgro

April 4, 2017

Contents

1 Introduction

From the MODELLER web site[1] :

`MODELLER` is used for homology or comparative modeling of protein three-dimensional structures (Webb and Sali 2016, Marti-Renom et al. (2000))

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** (Sali and Blundell 1993, Fiser, Do, and Sali (2000)), and can perform many additional tasks, including *de novo* modeling of loops in protein structures, optimization of various models of protein structure [. . .]

Figure 1.

[1] https://salilab.org/modeller/

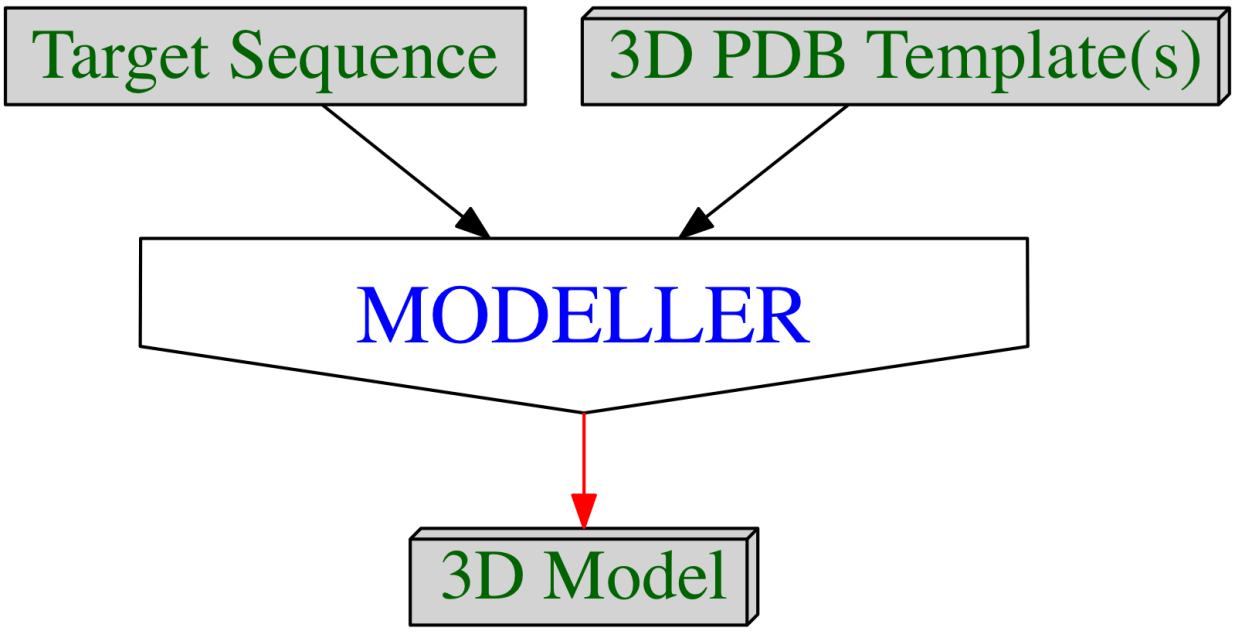

Figure 1: MODELLER process flow.

The current release of Modeller is `9.18`, which was released on Feb 22nd, 2017 and is intalled on all the iMacs. However, each user should register with the web site to obtain the install keyword at https://salilab.org/modeller/registration.html

2 Acknowledgments

Part of this tutorial is from "Comparative Protein Structure Prediction MODELLER tutorial" by Marc A. Marti-Renom (PDF)[2]

3 Set-up

We will use `MODELLER` on a Macintosh system but it would work exactly the same on other platforms.

`MODELLER` is made of a collection of python scripts, that the user just has to modify to reflect the name of the target sequence(s) and the template structure(s).

It is always good practise to create a directory for a specific project. Let's create a directory on the desktop called `MOD1` where we will save the necessary files.

TASK

Create a folder/directory on your desktop called `MOD1` or any name you wish.

3.1 Terminal

Then `MODELLER` is invoked on the line command with the name of the current version. The current release is `9.18` and is invoked on the line command as **`mod9.18`** followed by the name of the script to run.

[2]http://sgt.cnag.cat/www/presentations/files/slides/20081104_MODELLER_Tutorial.pdf

TASK

Open a text `Terminal`.

It is necessary to open a text `Terminal` to run `MODELLER`. On Mac `Terminal` is found as `/Applications/Utilities/Terminal` but can easily be launched by typing `Terminal` within the "Spotlight Search" on the top-right corner of the Mac screen (magnifying glass icon.)

(On a Windows computer you would need to open a command line by searching for the `cmd` program with Cortna or the Start button.)

Next it is necessary to change where the `Terminal` is "looking" with the "change directory" `cd` command:

```
cd MOD1
```

In the next section we will add files and scripts to this folder.

3.2 Text editing

Script and/or plain text files can be edited on a Macintosh with the **`TextEdit`** built-in text editor. However, it is necessary to verify that the format is plain text by engaging the menu **`Format` > `Make Plain Text`** if the program opens in `Rich Text` format as it is often the default behavior.

The full screen word processor **`nano`** could also be used from the `Terminal` also available on Linux systems.

Windows users can use `Notepad` or `Wordpad` to easily create plain text files.

4 Using MODELLER

To run `MODELLER` we need input data: sequence(s) and 3D template(s) in the proper format as well as python scripts. The later are found on the `MODELLER` web site as example files to be modified.

The output will consist of 1 or more (if requested in the script) 3D PDB format models, an alignment of sequence(s), a log file and other ancilary output.

INPUT:

- sequence(s) target(s). FASTA/PIR format
- structure(s) template(s). PDB format
- Python command file(s)

OUTPUT:

- Target-Template Alignment
- Model in PDB format
- Other data

5 Simple example

This simple example assumes that some prior study work has been done on the sequence to be modeled to find a suitable 3D template.

The purpose of the exercise is to create a 3D model from the sequence of the "brain lipid-binding protein" (blbp) based on one existing 3D structure with a different sequence that has been solved and published on the Protein Data Bank (PDB) (Berman et al. 2000).

Prior analysis reveals that the sequence of the "brain lipid-binding protein" is closely related of that of "human muscle fatty acid binding protein" that has been solved by X-ray crystallography with accession code `1HMS` 1hms.pdb (Young et al. 1994).

The protein sequences are 62% identical and 78% similar with no sequence gap as revealed by a simple 2 sequence BLAST alignment. Therefore these are a perfect subject for homology modeling.

Score	Expect	Method	Identities	Positives	Gaps
177 bits(450)	8e-64	Compositional matrix adjust.	81/130(62%)	102/130(78%)	0/130(0%)

```
Query  1    VDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKNT  60
            VDAF  TWKL DS+NFD+YMK+LGVGFATRQV ++TKPT II + G  + ++T  TFKNT
Sbjct  1    VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNT  60

Query  61   EISFKLGVEFDETTADDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHG  120
            EI+F+LG EF+ET+ DDR  KS+V LDG KL+H+QKWDG+ET   RE+ DGK+++TLT G
Sbjct  61   EINFQLGEEFEETSIDDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFG  120

Query  121  TAVCTRTYEK  130
              V  R YEK
Sbjct  121  DIVAVRCYEK  130
```

5.1 INPUT: Target sequence

TASK

Create a text file called `blbp.seq` containing the sequence sequence in the `MOD1` directory.

Example Target: Brain lipid-binding protein (BLBP).

BLBP sequence in PIR (`MODELLER`) format:

```
>P1;blbp
sequence:blbp::::::::
VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSIDDRNCKSVV
RLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

5.2 INPUT: download PDB structure

The input structure has accession code `1HMS`.

The downloaded file will appear in your `Downloads` directory as 1HMS.pdb.

TASK

Move downloaded file `1HMS.pdb` to the `MOD1` directory.

5.3 INPUT: Align sequences

The target sequence and 3D structure sequence need to be aligned and saved in a file with the proper format.

To accomplish this we need to edit a python script containing the name of the files containing the sequences. The sequence from the PDB file will be extracted from the PDB file itself by the script and `MODELLER`.

TASK

Create a text file called `align.py` with the following content and save it in folder `MOD1`:

```
# Example for: alignment.align()
# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()
aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))
# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

> *Note*: Since these are python functions, they need parentheses `()` even if there is nothing inside them. The meaning of the commands can be found under `MODELLER` online manual https://salilab.org/modeller/manual/

Here is a detail for this script:

- `log.verbose()` : display all log output
- `env = environ()` : create a short name for `environ()`
- `environ()` : contains most information about the MODELLER environment, such as the energy function and parameter and topology libraries [...].
- `aln = alignment(env)` : This creates a new `alignment` object; by default, this contains no sequences. `aln` is the short name for this object.
- `mdl = model(env, file='1hms')` : create a new 3D model. Here we pass on the information about the PDB file and atom information will be read. `mdl` is the short name for this object.
- `aln.append_model(mdl, align_codes='1hms')` : append the sequence of `1hms` to the alignment. In more complex analyzes there could be multiple PDB codes passed on.
- `aln.append(file='blbp.seq', align_codes=('blbp'))` : append the target sequence to the alignment.
- `# The as1.sim.mat similarity matrix is used by default:` This is a comment line
- `aln.align(gap_penalties_1d=(-600, -400))` the command `aln.align` create the alignment based on the indicated gap penalties.
- `aln.write(file='blbp-1hms.ali', alignment_format='PIR')` the alignment is written in `PIR` format.
- `aln.write(file='blbp-1hms.pap', alignment_format='PAP')` the alignment is written in `PAP` format.

It is worth noting the following point:

- the codes are within single quotes, for example `'1hms'`
- If there are multiple arguments passed to a function, there is a space after the comma `,` for example before the word `alignment_format=` in the lines above.

5.3.1 Run script to create alignment files

br>

TASK

Run alignment script `align.py` within `MOD1`.

Verify that you are within the MOD1 directory:

```
pwd
```

The answer should be something like:

```
/Users/yourname/Desktop/MOD1
```

Now run the alignment script by typing:

```
mod9.18 align.py
```

This will create the files: blbp-1hms.ali, blbp-1hms.pap, and align.log.

To see the content of the alignment files we can use the simple cat command on the Terminal (or use the graphical interface with TextEdit for example.)

```
cat blbp-1hms.ali
```

```
>P1;1hms
structureX:1hms:   1 :A:+131 :A:MOL_ID  1; MOLECULE  MUSCLE FATTY ACID BINDING PROTEIN; CHAIN  A; ENGINE
VDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKNTEISFKLGVEFDETTA
DDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTYEKE*

>P1;blbp
sequence:blbp:     : :     : :::-1.00:-1.00
VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSI
DDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

This alignment extracted sequence information from the PDB file for 1HMS including header information about the content that is placed within the header of structureX:1hms.

The .ali formatted alignment file is used later by MODELLER to create the 3D model(s).

The .pap formatted alignment is easier for human eyes to evaluate the alignment with the marked conserved (identity) regions.

```
cat blbp-1hms.pap
```

```
 _aln.pos         10        20        30        40        50        60
1hms      VDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKNTEISFKLGV
blbp      VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGE
 _consrvd ****  **** ** *** *** **********   **** **   *      *  ******* * **

 _aln.p   70        80        90       100       110       120       130
1hms      EFDETTADDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTYEKE
blbp      EFEETSIDDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA
 _consrvd ** **  ***  ** * *** ** * ***** **   **  ***   *** *  *  * ***
```

5.4 Model building

We now have the necessary "ingredients" to create the 3D model:

- aligned sequences
- 3D original template

We now need to create/edit the MODELLER python script that will list these ingredients and call the MODELLER functions to build the model.

TASK

Create a text file called `model.py` with the following content and save it in folder `MOD1`. Note that the comments noted with `#` do not need to be re-typed if not creating the file with a copy/paste method. The blank lines are only for text clarity and can also be omitted if desired.

To create the file you can use `TextEdit` or `nano` for example.

```
# Homology modelling by the automodel class
from modeller.automodel import *  # Load the automodel class
log.verbose()                     # request verbose output
env = environ()                   # create a new MODELLER environment

a = automodel(env,
   alnfile = 'blbp-1hms.ali',     # alignment filename
   knowns = '1hms',               # codes of the templates
   sequence = 'blbp')             # code of the target

a.starting_model= 1               # index of the first model
a.ending_model = 1                # index of the last model
                                  # (determines how many models to calculate)

a.make()                          # do the actual homology modelling
```

Remarks: The `automodel` function is renamed `a` and the "dot notation" is used to call on sub function appended to `a` as it is the usual writing mode in python.

In this simple file we create only one model, but to obtain *e.g.* 5 models the `a.ending_model` argument would be set to `5`.

5.5 Run model building script

TASK

Run `model.py` within `MOD1` in the same manner as we ran the `align.py` script:

```
mod9.18 model.py
```

This will create the following files:

```
blbp.B99990001.pdb
blbp.D00000001
blbp.V99990001
model.log
blbp.ini
blbp.rsr
blbp.sch
```

The final 3D model is called `blbp.B99990001.pdb` and that is the "end product" that was desired.

In real life, multiple models would be calculated (*e.g.* `5`) and various evaluation methods could be applied to decide which are "best."

You can explore the content of the remaining file (all text files) with the `less -S` command that will display the file content to the screen without wrapping long lines.

6 Compare model and template graphically

Now that we have a model we can compare the structure onbtained with the original template.

For this you can use `Chimera` or `PyMOL` or any other molecular graphics software that can read `PDB` files.

6.1 PyMOL

To open and compare files in PyMOL open the PyMOL program first.

- At the line command type: `fetch 1hms` to load the original template file.
- Using the menu cascade **`File > Open...`** navigate to the `MOD1` directory to open file `blbp.B99990001.pdb`.
- Use left mouse button to rotate structure.

Note: the 2 structures will not be superimposed at first and it will be necessary to align them in 3D.

- Align the structures: on the Names panel at right, click on **`A`** (action) button next to the line that reads **blbp.B99990001.pdb 1** for the model. Following further down on this pull-down menu follow the menu cascade: **`A > align > to molecule (*/CA) > 1hms`**

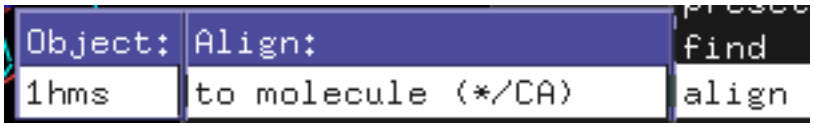

Figure 2: "***Align structures menu.***"

Figure 3: "***Open and align structures in PyMOL.***"

- To hide or show either structure simply click **once** on the name of the structure on the list at the right hand side Names panel.
- In order to highlight the bound lipid use the following menu casade next to the **all** line on the right hand side: **`all > S > organic > sticks`**
- To hide the red dot water molecules: **`all > H > waters`**

  Note: only the protein is modeled, the ligands are not modeled by `MODELLER`. These are typically written as `HETATM` within the PDB file.

6.2 Chimera

If you prefer using Chimera:

- Open Chimera
- Open template structure: `File > Fetch by ID...` and enter `1HMS` in the `Fetch Structure by ID` in the text space next to the `PDB` button. This will open the structure in "first view" mode as a cartoon ribbon diagram.
- Open the model: **`File > Open...` and navigate to the `MOD1` directory to open file `blbp.B99990001.pdb`. The default view will also be as a cartoon ribbon.

Note: the 2 structures will not be superimposed at first and it will be necessary to align them in 3D.

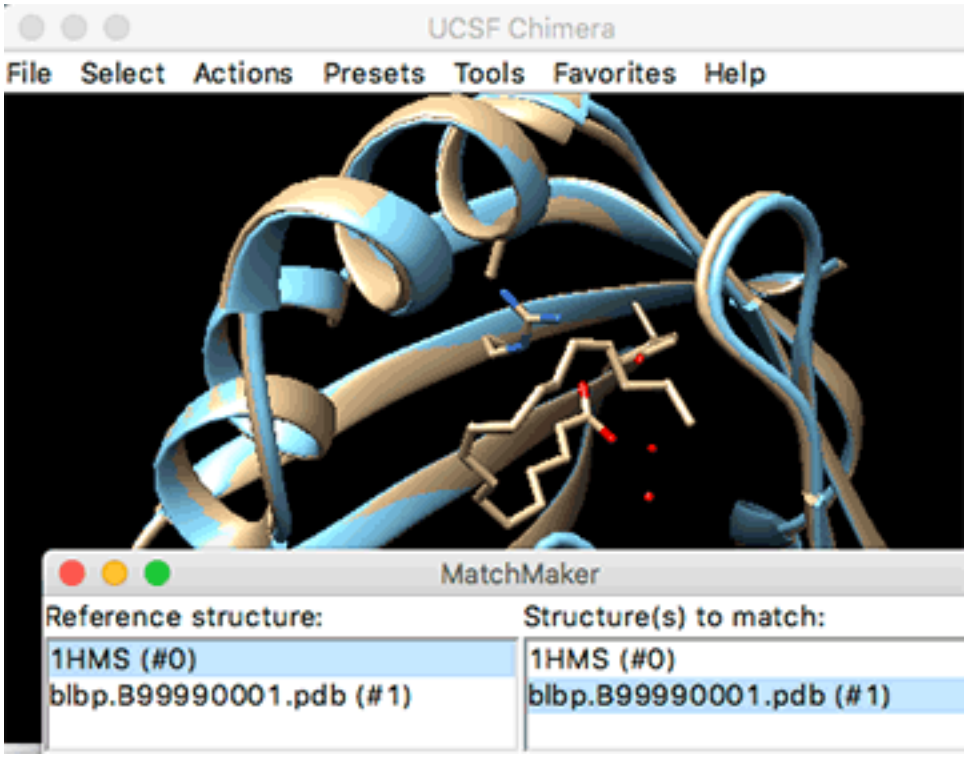

Figure 4: "***Open and align structures in Chimera.***"

- `Tools > Sequence Comparison > MatchMaker` will open the `MatchMaker` window. Keep everything the the current default and click `1HMS (#0)` for the "Reference structure" and `blbp.B99990001.pdb (#1)` for the "Structure(s) to match"
- Click `Apply` and the 2 structures will be aligned.
- Use left mouse button to rotate structure.

7 MODELLER tutorials online

7.1 Official web site

The `MODELLER` web site offers tutorials with different levels of difficulty https://salilab.org/modeller/tutorial/ :

1. Basic Modeling. *Model a sequence with high identity to a template.* This exercise introduces the use of MODELLER in a simple case where the template selection and target-template alignments are not a problem.
2. Advanced Modeling. *Model a sequence based on multiple templates and bound to a ligand.* This exercise introduces the use of multiple templates, ligands and loop refinement in the process of model building with MODELLER.
3. Iterative Modeling. *Increase the accuracy of the modeling exercise by iterating the 4 step process.* This exercise introduces the concept of MOULDING to improve the accuracy of comparative models.
4. Difficult Modeling. Model a sequence based on a low identity to a template. This exercise uses resources external to MODELLER in order to select a template for a difficult case of protein structure prediction.
5. Modeling with cryo-EM. Model a sequence using both template and cryo-EM data. This exercise assesses the quality of generated models and loops by rigid fitting into cryo-EM maps, and improves them with flexible EM fitting.

7.2 Other courses

Virtual Proteomics Laboratory - Experiment 10: Homology Modelling - http://iitb.vlab.co.in/?sub=41&brch=118&sim=657&cnt=2

REFERENCES

Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. 2000. "The Protein Data Bank." *Nucleic Acids Res.* 28 (1): 235–42.

Fiser, A., R. K. Do, and A. Sali. 2000. "Modeling of loops in protein structures." *Protein Sci.* 9 (9): 1753–73.

Marti-Renom, M. A., A. C. Stuart, A. Fiser, R. Sanchez, F. Melo, and A. Sali. 2000. "Comparative protein structure modeling of genes and genomes." *Annu Rev Biophys Biomol Struct* 29: 291–325.

Sali, A., and T. L. Blundell. 1993. "Comparative protein modelling by satisfaction of spatial restraints." *J. Mol. Biol.* 234 (3): 779–815.

Webb, B., and A. Sali. 2016. "Comparative Protein Structure Modeling Using MODELLER." *Curr Protoc Bioinformatics* 54 (June): 1–5.

Young, A. C., G. Scapin, A. Kromminga, S. B. Patel, J. H. Veerkamp, and J. C. Sacchettini. 1994. "Structural studies on human muscle fatty acid binding protein at 1.4 A resolution: binding interactions with three C18 fatty acids." *Structure* 2 (6): 523–34.