Homology Modeling

- Practical course -

Introduction

Three main methods are used to determine 3D structure of a protein from its sequence: homology modeling, threading and *de novo* methods. De novo and threading methods are used when no homologuous structure is available (low identity percentage), but these methods are not yet very accurate. The most popular *de novo* method is **`Rosetta`**[1], which performs the best at this time. A web server implementation of the method called **`Robetta`** is available at http://robetta.bakerlab.org/

Homology modeling is an improved method based on the fact that homologous proteins have similar 3D structures. In the case that a homologue of the protein of interest is available, with such tools as **`MODELLER`**, it's possible to build a model from the template 3D coordinates and an alignment of amino-acids sequences. **`MODELLER`** applies the structure of the template to the protein of interest taking into account the sequence constraints (steric clashes, electrostatic interactions, amino acids secondary structure propensities, etc).

Briefly, models generation consists of four steps. First, scan sequences databases for sequences similar to our target. One can consider a sequence as a template candidate, if the E-value of the alignment is lower than 10^{-4}; second, align the template with the target ; third, automatically generate models with template coordinates and an alignment ; finally, check generated models validity.

Nota bene: the E-value (Expected value) is the probability that the obtained score is due to hazard, ie the alignment is not significant.

Goal

The main goal of this practical is to introduce you the essential tools for homology modeling with a classical example : the *calmodulin*. The advantage of this protein is that 3D structure has been solved, so, at the end of this practical, you will be able to compare your models with the real structure.

Remarks

All paths are relative to the `PracticalRessources/HM` path. Don't hesitate to ask instructors for more informations. Don't worry if you don't finish this practical, everything can be done using web servers, so you will be able to finish it when you want, where you want, independently of the operating system you are using. If you have any questions about this practical, please contact: Gaëlle Debret (debret@ebgm.jussieu.fr) and Julien Maupetit (maupetit@ebgm.jussieu.fr).

[1] **Simons KT, Kooperberg C, Huang E, Baker D.**, *J Mol Biol.* 1997 Apr 25;268(1):209-25

I. *Search for a similar sequences in databases*

The sequence file `unknown.seq` (in the `sequences` directory) contains the sequence of the unknown structure to model. First we have to search for the family of this protein with the **`BLAST`** program. This will guide us through the choice of a template structure for modeling.

Launch a web browser from your terminal (type `firefox` for example), and go to the NPS@[2] server from the IBCP[3] at the following address http://npsa-pbil.ibcp.fr/. Click on the link « *work with your own database* », and then upload your database file `693331.dbextr` located in the `db` subdirectory. Note that, since the structure of this protein is known, its sequence is in all disposable databases ; this explain the definition of your own working database.

When your file has been completely uploaded, then start a **`BLAST`** search of your sequence in the defined database.

The solution is available in the `results/blast/` directory.

`BLAST` results analysis:

- *Search for sequences with a known 3D structure. How could we know it by reading lines of the output listing ?*
- *According to you, what could be the best template ?*
- *The selected template is `1cdl`. This is not the only solution. We could have chosen `1ahr`, `1qiw`, `1lin`, `1g4y` or `1ag2` which have a best score. Why don't we choose these templates ?*
- *`1dmo` is following `1cdl`, why don't we choose this one ?*
- *Regarding the E-values is the choice of `1cdl` valid ?*

II. *Alignments*

Now we have selected the template to build the model, we need to align the unknown sequence with the template sequence. You can use **`CLUSTALW`** to do so. An implementation is available at the following address: http://npsa-pbil.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_clustalw.html (Note that you are free to use another server like the EBI, http://www.ebi.ac.uk/clustalw/, or to download and install **`CLUSTALW`** from the EBI).

`CLUSTALW` needs concatenated sequences in fasta format as input to multiple alignment. Note that sequences in fasta format are in the `sequences` directory (`.fast` extensions), but **we recommend you to find them by yourselves on the web**.

Submitted data looks like:

```
>sequence1
ATAREFAREFEF...
>sequence2
EAZDAZDRGER...
```

[2] Network Protein Sequence Analysis

[3] *Institut de Biologie et Chimie des Protéines*

Align the unknown sequence with the `1cdl A` chain sequence first, and then with the `1avs A` chain sequence.

The solution is available in the `results/clustalw/` directory.

`CLUSTALW` results analyze:

- *Compare the identity rates between the two alignments.*
- *Why did we choose the A chain of `1cdl` ?*
- *What can you say about the alignment and the missing parts ('-' or 'X') ?*
- *Change **`CLUSTALW`** parameters in the form. What is evolving on the alignment ?*

III. *Automatic modeling*

We use the **`MODELLER`**[4] program available at http://salilab.org/modeller/. It is installed on your computers. This program, used to generate models, has its own particular syntax ; please pay attention to this point. A knowledge of the Python language should help you with this step, but is not essential.

`MODELLER` needs an alignment as input with `.ali` extension. An example of the file format is in the file `modeller/examples/alignment_example.ali`. You also need to give **`MODELLER`** instructions to generate your models in a `.py` script file. An example is in `modeller/examples/model-single.py`.

*Prepare your own `.ali` file with the previous **`CLUSTALW`** alignment. And fill the `.py` file to generate 5 models. The command syntax is : `mod8v2 myScript.py`.*

The solution of the input script is in `modeller/resources/alignement1cdl.py`.

The quality criteria of the model is obtained by the minimization of an objective function. The value of this score is presented in the second line of the PDB file (`REMARK` field).

Which one is the best model ?

5 models are presented in `modeller/resources/models/unknown.B9999000[1-5].pdb`

Analyze this 5 models and comment on them.

[4] Copyright © 1989-2006 Andrej Sali

IV. *Models validation*

Some methods can be used to check your models validity (for a complete listing please refer to http://salilab.org/bioinformatics_resources.shtml).

We invite you to use the meta-method **Eval123D** at: http://bioserv.cbs.cnrs.fr/HTML_BIO/frame_valid.html

Compare models scores and localize problematic regions. Which model is the best ?

V. *Model visualization*

Many molecular visualization programs are available. The most popular of them are **Rasmol**, **VMD** and **PyMol**. All of these programs are available on all platform (Windows, Linux, MacOsX). Each one of them has its own advantage : **Rasmol** is well working on low CPU frequency workstation and **PyMol** has a nice internal 3D renderer, and is highly extensible with Python scripts. Here, we use the **VMD**[5] program for its complete toolbox, and its simplicity.

First superpose the template PDB file 1cldA, and your best model, using the **iSuperpose** server : http://bioserv.rpbs.jussieu.fr/cgi-bin/iSuperpose and then visualize them with **VMD**.

This method is based on the **TMalign** program (3D structural alignment) and the **QbestFit** method(to fit the aligned regions best).

Then, discard the template and visualize the superposed known 3D structure with PDB code 4tnc.

What are your conclusions ? Do you have any hypotheses explaining our observations ?

[5] http://www.ks.uiuc.edu/Research/vmd/