Biology 5476
Homology Modelling Lab
November, 2008

Rob Yang
Marshall Lab
ryyang@artsci.wustl.edu

Introduction

As you learned this week, homology modelling is yet another powerful tool in the field of computational modelling. It offers a clever way of "guessing" the 3D structure from its target 1D sequence using existing structures of similar proteins. Consequently, accuracy depends on similarity. In the more challenging cases, it is sometimes possible to identify sub-domains that share high similarities even when the global sequence similarities are lacking. In addition to global structural prediction, it is also very powerful at constructing protein complexes based on related templates. So like every other computational modelling tools, be critical and careful in how and when you use it. We'll walk through the basic steps to perform both structural prediction and complex construction today.

The software we are going to use today is the "Modeller" package by Andrej Sali at UCSF (http://www.salilab.org/modeller/). In the interest of time, the exercise today is meant to get your feet wet, so you'll have a place to start if you do need to use it in your own research down the road. Just remember, "it's running" is not the same as "it's working", the devil is always in the details so the manual is your best friend if you want to avoid the classic garbage-in garbage-out situation.

Part I

Problem setup: you are given a sequence of a protein, ErbB2, construct a homology model based on the structure of its related protein, EGFR.

1) Copy input files from ~ryang/BIO5476_homology_2008/part1/ to your own directory.

Ex: cp ~ryang/BIO5476_homology_2008/part1/* ./

2) Open up the erbb2.aln file using your favorite text editor, this is your ErbB2 sequence file.

Ex: /Applications/TextEdit.app/Contents/MacOS/TextEdit erbb2.aln

3) Display 1IVO.pdb using your favorite visualization program, this is your EGFR structural file.

Ex: /Applications/MacPyMOL.app/Contents/MacOS/MacPyMOL 1IVO.pdb

Notice that there are multiple chains: chain A and B represent 2 EGFR monomers, C and D are another protein (EGF) bound to each EGFR monomers. For our purpose, chain A is the template structure used to construct ErbB2, so ignore the other chains for now.

4) Now, we are going to align the sequence of ErbB2 with that of EGFR. There are many ways to achieve this, but we are going to do it in Modeller. Open the align2d.py script, the 3 lines before calling align2d() specify the alignment input files, and the 3 lines after specify the outputs.
Run the script:

mod9v4 align2d.py

Two result files are produced: erbb2-1IVOA.aln and erbb2-1IVOA.pap. These two files represent the same result: global pair-wise sequence alignment. The .pap file is for easy visualization by us, while the .aln file will be used by the subsequent steps. Note the high sequence similarity.

5) Next, we will perform the actual homology modelling using the sequence alignment. Open the model-single.py script, DOPE and GA341 are the built-in structural evaluation scoring functions: negative value=good for DOPE, 1.0=good for GA341. 10 models will be made.
Run the script:

mod9v4 model-single.py

The DOPE and GA341 scores of all the models are written at the end of the log file (last 2 columns). You can visually examine all 10 models and the original EGFR template.

Ex: /Applications/MacPyMOL.app/Contents/MacOS/MacPyMOL 1IVO.pdb erbb2.*.pdb

If you are using pymol, try aligning all the models to the template.

Q1: Where are some of the obvious deviations within the models? How are the sequence alignment for these regions? Contrast that with the regions that have good agreement among models. Write a few sentences on any trend or observations that you find interesting.

Q2: The ErbB2 structure has actually been solved (PDB 2A91). Load 2A91_clean.pdb to pymol and compare with the models. Are the models satisfactory?

Pick your favorite model (you can do this by the scores or just how your feel), substitute its full pdb name in place of “erbb2.B99990006.pdb” in the evaluate_model.py. Run the script. Plot the profiles using gnuplot:

gnuplot
>plot "1IVO_A.profile" using 1:42 with lines, "erbb2.profile" using 1:42 with lines, “your_model_filename.profile” using 1:42 with lines

This further breaks down the model for you to suggest the well-defined regions (low) and disordered regions (low) within the given model. The 1IVO_A and 2A91_clean files serve as your “controls”.

Part II

Problem setup: can we predict a EGFR-ErbB2 heterodimer based on the EGFR homodimer structure?

Crystallization of heterodimers usually presents a much more difficult challenge experimentally, especially when the monomers are capable of homodimerize and hence resulting in heterogenous populations of dimers. This is why homology models of the heterodimer can serve as a good hypothesis generator about important residue-residue interactions.

1) Copy files from ~ryang/BIO5476_homology_2008/part2/ to your directory. The "heterodimer-homodimer.aln" file is the alignment of EGFR-EGFR homodimers and 2A91-EGFR heterodimers. The "homodimer.pdb" file is consisted of chain A + B + D from 1IVO. "Heterodimer.pdb" file is constructed by superposing 2A91 onto monomer A of 1IVO.

2) Run the refine.py script

Q3: Comment briefly on the difference between the superposed initial structure and the modeller derived models. Now try different initial heterodimer configurations by constructing a file that has your favorite model from Part I.

Extra (only if you want to practice)
ErbB3 is another protein related to both EGFR and ErbB2. Copy the ErbB3 pdb file from ~ryang/BIO5476_homolog_2008/extra/ to your directory. Repeat part I using ErbB3 as the template. Make sure you visualize EGFR and ErbB3 side by side. The punch line of this exercise is that even when sequences are highly similar, you can never ever discount conformational change when you model. Now construct a ErbB2-ErbB3 heterodimer model. (Hint, you will have to model the ErbB3 monomer to a different conformation first).