Project No.6 - Comparative Modeling of Protein Structure

Taylor Kessinger, Zarin Shakibaei, Fabio Zanini

18.06.12

Abstract

Modeller is a popular software tool for predicting protein structures using homology modelingin this case, by satisfaction of spatial restraints derived from similar protein structures. We use Modeller to predict the structure of a protein sequence (PDB ID: `1yje`) by comparing it to a number of protein sequences with known structures. Sequence alignment and structure prediction can both be performed using built-in Modeller classes. We do so, then visualize the result using PyMol. The predicted structure is reasonably close to the true structure of the target.

Introduction

The particular sequence of amino acids in a protein is not necessarily important except insofar as it determines the fold of a protein; proteins are thus more robust to changes at the sequence level than at the structural level. This yields the evolutionary result that protein structures are much more conserved than protein sequences. Homology modeling exploits this fact by fitting a desired protein sequence with a potentially unknown structure (the target) to any number of similar sequences with known structures (the templates). Given a good alignment between a target and template sequence, a very accurate structure can be predicted. Homology modeling is obviously most successful when the target and template have similar structures.

The goal of this project is to predict the structure of a target, `1yje:A`, by using the Python homology modeling package, Modeller. As it happens, `1yje:A` has a known structure in the Protein Database (PDB) [4], so it is possible to check the predicted structure against the experimentally determined one.

Materials and Methods:

Modeller overview

Modeller is a Python package that performs homology modeling by satisfaction of spatial restraints. The basic procedure is very simple:

- The target sequence is aligned against one or more template structures using the sequences and the .pdb file for the templates.
- Spatial features from the templates, such as dihedral angles, C_α-C_α distances, and hydrogen bonds, are determined from the structures of the templates. These are used to derive spatial restraints on the target structure. [6]
- A model for the target structure is determined by satisfying these restraints.

One example of a "spatial restraint" is described in [6]: If, say, a hydrogen bond to the main chain at a certain residue appears in all or many comparable template structures, it is reasonable to assume that it also appears in the target. In the original implementation of Modeller, "reasonable to assume" means that a high probability is placed on this feature appearing in the target structure; probability density functions constraining stereochemistry, main- and side-chain conformations, amd so on are constructed, then combined into a single molecular pdf. Optimizing with respect to this pdf yields the most probable structure given the alignment. The conjugate gradient method is employed by Modeller for this optimization. Here, rather than rely on the molecular pdf, we employ a statistical potential optimized for model assessment, namely DOPE (Discrete Optimized Protein Energy). [7]

Initial search

CS-BLAST, or "context-specific BLAST" [2], an improved version of the Basic Local Alignment Search Tool [1] that takes into account the context of neighboring amino acids in a sequence, was used to find initial matches. Due to the use of context information, CS-BLAST finds a higher number of remotely homologous sequences than BLAST does at a given error rate. A number of hits were returned, namely:

- The target sequence from the PDB.
- A number of hits with $e < 0.01$ with reasonable chunks in common with the target sequence.
- A number of hits with only a domain in common in the middle of the sequence or at the N-terminus. A large gap in e-value was visible between the previous hits and these.

Next, MUSCLE (MUltiple Sequence Comparison by Log-Expectation) [3] was used to generate an initial multiple sequence alignment using the good ($e < 0.01$) matches. Six iterations of muscle took about three seconds, meaning the sequences were essentially already aligned.

Input data and alignment

Unfortunately, FASTA files are inappropriate for use with Modeller. This is for two reasons. The first is obvious: Modeller is a homology modeling package, so some information about the structure of template proteins is needed to generate a model. The second is less obvious: A FASTA alignment file does not contain information specifying which positions of the alignment correspond to which residues of the protein. To remedy this, Modeller's modeling capabilities require that input be provided in PIR (Protein Information Resource) format, which contains not only the aligned residues, but an extra comment line specifying the start and end position of the alignment.

To this end, Modeller also contains its own alignment class. Modeller's aligning capabilities include several advantages; for example, gap penalties are lessened outside of secondary structures, based on the argument that secondary structures should be much less robust to insertions and deletions than unstructured parts of the protein. FASTA files can be converted into PIR format naïvely, with the appropriate comment line added; this and the PDB files for the template sequences are the minimum requirements needed to generate an alignment. The `alignment.align2d()` function allows for a single sequence alignment; the `alignment.salign()` function enables a multiple sequence alignment. Both can save an alignment file in PIR format.

Scripting

Modeller is a command-line only tool, and has no graphical user interface; instead, a script file containing Modeller commands should be provided. Once

installed, Modeller can be called from the command line with the `mod9.10` command, which automatically calls Python and imports the necessary `modeller` module. The name of the desired script can be provided as an argument. Standard output is suppressed and saved to a .log file.

Our program consists of several pieces of Python code. A brief outline is provided below.

- `parse_blast_results.py` contains the function `parse_blast_results`, which parses a FASTA file corresponding to our initial alignment and saves the sequence and chain IDs.
- `read_templates.py` contains two functions; `read_templates()`, which reads sequence and chain IDs and stores them in a dict, and `get_tplname()`, which returns a typical alignment code name for a template.
- `model_align.py` is a simple test file. It creates a `environ()` object, which is needed for all MODELLER scripts. It then aligns the target, `1yje:A`, against a random template, `3tx7:B`. The `automodel` class is invoked to generate a model pdb file.
- `align_model_all.py` is an improvement on `model_align.py` that produces single alignments of the target against a range of templates, then performs the corresponding predictions.
- `model_align_multiple.py` generates an MSA of the target and the top three hits (those that are not simply copies of the target structure). It then invokes the `automodel` class, which assigns weights to the various templates based on their sequence identity with the target. The `assess_methods` option allows DOPE scores to be output to the .log file.
- `plot_figures_pymol_multiple.py`, which incorporates the Python structure visualization module `PyMol` to visualize a .pdb file. It must actually be called from within PyMol and has options to plot the predicted structure against template structures or the true structure, then save either to file.

The `automodel` class offers a wide range of options for incorporating water, disulfide bridges, and a variety of other features not incorporated by default, as well as refining only parts of the model. For our purposes, however, the default options sufficed.

Visualization

Our PyMol script enables us to visualize the .pdb format outputs of our alignment and prediction script and compare it against the known structure of `1yje:A`. As a sanity check, we can also predict the structure of the target using itself as a template (since `1yje:A` has a structure in the PDB). Obviously, we

```
EXPDTA    THEORETICAL MODEL, MODELLER 9.10 2012/06/15 15:44:24
REMARK   6 MODELLER OBJECTIVE FUNCTION:     24258.8906
REMARK   6 MODELLER BEST TEMPLATE % SEQ ID: 100.000
ATOM      1  N   ASN     1     -30.397 -13.376 -15.958  1.00113.58           N
ATOM      2  CA  ASN     1     -29.728 -12.136 -15.500  1.00113.58           C
ATOM      3  CB  ASN     1     -28.599 -12.449 -14.502  1.00113.58           C
ATOM      4  CG  ASN     1     -29.218 -12.875 -13.182  1.00113.58           C
ATOM      5  OD1 ASN     1     -30.324 -12.461 -12.841  1.00113.58           O
ATOM      6  ND2 ASN     1     -28.482 -13.719 -12.409  1.00113.58           N
ATOM      7  C   ASN     1     -29.117 -11.362 -16.630  1.00113.58           C
ATOM      8  O   ASN     1     -28.989 -11.856 -17.750  1.00113.58           O
ATOM      9  N   LEU     2     -28.735 -10.104 -16.337  1.00 76.93           N
```

Fig. 1: Sample of a PDB file corresponding to `1yje:A`, constructed with 15 templates.

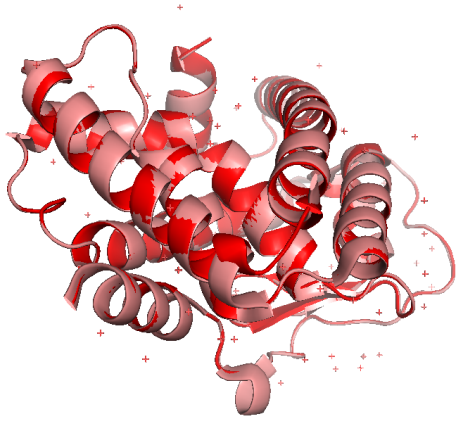

Fig. 2: The target threaded onto itself. Colors: red = our model, salmon = true structure.

expect to see a perfect match in that case, which we do, as can be seen by examining Fig. 2. This confirms that homology modeling is, at least *prima facie*, a sensible approach, and that Modeller does what it claims to do.

We also show the results of threading the target onto the top two, top five, and top ten similar PDB entries, as ranked by *e*-value, in Fig. 3, Fig. 4, and Fig. 5. A number of options are available for evaluating the quality of protein models [5]. Our structures are each selected by generating three multi-template models and selecting the one with the top DOPE score. Notably, introducing the additional entries does not seem to improve the quality of the prediction that much. Even with only two templates, the quality is very high, howeveronly a few small flaws, such as the slightly wrong start position for a sheet or helix,

are visible.

Fig. 3: A multiple-sequence model: Our unknown sequence threaded onto two similar PDB entries. Left: The predicted structure against the true one. Right: The same against all ten models. Colors: red = our model, green = true structure, other = templates.

Fig. 4: A multiple-sequence model: Our unknown sequence threaded onto five similar PDB entries, including ones with lower sequence identity to our target. Left: The predicted structure against the true one. Right: The same against all five models. Colors: red = our model, fuchsia = true structure, other = templates.

In each case, DOPE scores on the order of -30k result, confirming the accuracy of our predictions. Furthermore, the accuracy can be seen visually.

Discussion:

Homology modeling is a powerful tool for predicting the structure of unknown proteins. In fact, it has been suggested that the PDB is essentially complete enough to "solve" the structure prediction problem, with the only limiting factor being inadequacies in sequence alignment [8]. This problem, too, may soon be fixed, as tools like CS-BLAST continue to improve the quality and number of hits than can be acquired. Here, we have successfully demonstrated the accuracy of homology modeling when known template structures are available.

Fig. 5: A multiple-sequence model: Our unknown sequence threaded onto ten similar PDB entries. Left: The predicted structure against the true one. Right: The same against all ten models. Colors: red = our model, purple = true structure, other = templates.

One obvious shortcoming of homology modeling is its inability to model proteins that have essentially no homologues. Examples include the venoms of many animals, which are relatively young proteins that are very specific to that particular animal. Since they do not have significant sequence identity with other proteins of known structure, and since structure identity decreases rapidly when sequence identity sinks below about thirty per cent, there is little hope for homology modeling to be effective for these particular proteins.

Conclusion:

We have verified the effectiveness of Modeller for predicting the structure of a protein with an unknown sequence. Importantly, the accuracy of our prediction seems to be not terribly sensitive to the number of templates used; even a mere two templates is sufficient enough to generate a very accurate structure, as evaluated both by DOPE score and by visual inspection.

The development version of the source code for this project can be found at https://github.com/tkessinger/bi2-ss12-p6.

Bibliography

[1] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403 – 410, 1990.

[2] A. Biegert and J. Söding. Sequence context-specific profiles for homology searching. *Proceedings of the National Academy of Sciences*, 106(10):3770–3775, 2009.

[3] R.C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.

[4] Ralf Flaig, Holger Greschik, Carole Peluso-Iltis, and Dino Moras. Structural basis for the cell-specific activities of the ngfi-b and the nurr1 ligand-binding domain. *Journal of Biological Chemistry*, 280(19):19250–19258, 2005.

[5] Andriy Kryshtafovych and Krzysztof Fidelis. Protein structure prediction and model quality assessment. *Drug Discovery Today*, 14(78):386 – 393, 2009.

[6] A. Sali and TL Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Protein Structure by Distance Analysis*, 64:C86, 1994.

[7] Min-yi Shen and Andrej Sali. Statistical potential for assessment and prediction of protein structures. *Protein Science*, 15(11):2507–2524, 2006.

[8] Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current pdb library. *Proceedings of the National Academy of Sciences of the United States of America*, 102(4):1029, 2005.