Protein Structure Modeling With MODELLER

Narayanan Eswar[$], David Eramian, Ben Webb, Min-Yi Shen and Andrej Sali[$]

Departments of Biopharmaceutical Sciences and Pharmaceutical Chemistry,
and California Institute for Quantitative Biomedical Research,
University of California at San Francisco.

[$]Corresponding authors:

Byers Hall, Room 503B,

University of California at San Francisco,

1700 4th Street, San Francisco, CA 94158-2330, USA.

tel: +1 415 514 4227; fax: +1 415 514 4231;

e-mail: eswar@salilab.org; sali@salilab.org; web: http://www.salilab.org.

Running title: Protein Structure Modelling

Abstract

Genome sequencing projects have resulted in a rapid increase in the number of known protein sequences. In contrast, only about one-hundredth of these sequences have been characterized using experimental structure determination methods. Computational protein structure modeling techniques have the potential to bridge this sequence-structure gap. In the following chapter, we present an example that illustrates the use of MODELLER to construct a comparative model for a protein with unknown structure. Automation of similar protocols has resulted in models of useful accuracy for domains in more than half of all known protein sequences.

Key Words: Comparative modeling, fold assignment, sequence-structure alignment, model assessment, multiple templates, consensus modeling.

1. Introduction

The function of a protein is determined by its sequence and its three-dimensional (3D) structure. Large-scale genome sequencing projects are providing researchers with millions of protein sequences, from various organisms, at an unprecedented pace. However, the rate of experimental structural characterization of these sequences is limited by the cost, time, and experimental challenges inherent in the structural determination by x-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy.

In the absence of experimentally determined structures, computationally derived protein structure models are often valuable for generating testable hypotheses (***1***). Comparative protein structure modeling has been used to produce reliable structure models for at least one domain in more than half of all known sequences (***2***). Hence, computational approaches can provide structural information for two orders of magnitude more sequences than experimental methods, and are expected to be increasingly relied upon as the gap between the number of known sequences and the number of experimentally determined structures continues to widen.

Comparative modeling consists of four main steps (***3***) (**Fig. 1**): (i) fold assignment that identifies overall similarity between the target and at least one known template structure (see Section on Materials for definitions of these terms); (ii) alignment of the target sequence and the template(s); (iii) building a model based on the alignment with the chosen template(s); and (iv) predicting the accuracy of the model.

MODELLER is a computer program for comparative protein structure modeling (***4, 5***). In the simplest case, the input is an alignment of a sequence to be modeled with the template structure(s), the atomic coordinates of the template(s), and a simple script file. MODELLER then automatically calculates a model containing all non-hydrogen atoms, without any user intervention and within minutes on a desktop computer. Apart from model building, MODELLER can perform auxiliary tasks such as fold-assignment (***6***), alignment of two protein sequences or their profiles (***7, 8***), multiple alignment of protein sequences and/or structures (***9***), clustering of sequences and/or structures, and *ab initio* modeling of loops in protein structures (***4***).

MODELLER implements comparative protein structure modeling by satisfaction of spatial restraints that include (i) homology-derived restraints on the distances and dihedral angles in the target sequence, extracted from its alignment with the template structures (***5***), (ii) stereochemical restraints such as bond length and bond angle preferences, obtained from the CHARMM-22 molecular mechanics force-field (***10***), (iii) statistical preferences for dihedral angles and non-bonded inter-atomic distances, obtained from a representative set of known protein structures (***11, 12***), and (iv) optional manually curated restraints, such as those from NMR spectroscopy, rules of secondary structure packing, cross-linking experiments, fluorescence spectroscopy, image reconstruction from electron microscopy, site-directed mutagenesis, and intuition (**Fig. 1**). The spatial restraints, expressed as probability density functions, are combined into an objective function that is optimized by a combination of conjugate gradients and molecular dynamics with simulated annealing. This model building procedure is similar to structure determination by NMR spectroscopy.

In this chapter, we use a sequence with unknown structure to illustrate the use of various modules in MODELLER to perform the four steps of comparative modeling. This is followed by a Notes section that highlights several underlying practical issues.

2. Materials

2.1. *Hardware*

1. A computer running Linux/Unix, Apple Mac OS X, or Microsoft Windows 98/NT/2000/XP; 512 MB RAM or higher; about 100 MB of free hard-disk space for the software, example and output files; and a connection to the internet to download the MODELLER program and example files described in this chapter (*see* **Note 1**).

2.2. *Software*

1. The MODELLER 8v2 program, downloaded and installed from http://salilab.org/modeller/download_installation.html. Instructions for the installation are provided as part of the downloaded package; they are also available over the internet at http://salilab.org/modeller/release.html#install.

2. The files required to follow the example described in this chapter, downloaded and installed from http://salilab.org/modeller/tutorial/MMB06-example.tar.gz (Unix/Linux/MacOSX) or http://salilab.org/modeller/tutorial/MMB06-example.zip (Windows).

2.3. *Computer Skills*

1. MODELLER uses Python as its control language. All input scripts to MODELLER are, hence, Python scripts. While knowledge of Python is not necessary to run MODELLER, it can be useful to perform more advanced tasks.

2. MODELLER does not have a Graphical User Interface (GUI) and is run from the command-line by executing the input scripts; a basic knowledge of command-line skills on a computer is necessary to follow the protocol described in this chapter.

2.4. *Conventions followed in the text*

1. A sequence with unknown structure, for which a model is being calculated, is referred to as the “target”.

2. A “template” is an experimentally determined structure, and/or its sequence, used to derive spatial restraints for comparative modeling.

3. Names of files, objects, modules and commands to be executed are all shown in `monospaced` font.

4. Files with ‘`.ali`’ extensions contain the alignment of two or more sequences and/or structures. Files with ‘`.pir`’ extensions correspond to a collection of one or more unaligned sequences in the PIR format. Files with ‘`.pap`’ extensions contain an alignment in a user-friendly format with an additional line indicating identical aligned residues with a ‘*’. All input scripts to MODELLER are Python scripts with the ‘`.py`’ extension. Execution of these input scripts always produces a log file identified by the ‘`.log`’ extension.

5. A typical operation in MODELLER would consist of (i) preparing an input Python script, (ii) ensuring that all required files (sequences, structures, alignments, *etc.*) exist, (iii) executing the input script by typing `mod8v2 <input-script>`, and (iv) analyzing the output and log files.

3. Methods

The procedure for calculating a 3-dimensional model for a sequence with unknown structure will be illustrated using the following example: a novel gene for lactate dehydrogenase (LDH) was identified from the genomic sequence of *Trichomonas vaginalis*

(TvLDH). The corresponding protein had higher sequence similarity to the malate dehydrogenase of the same species (TvMDH) than to any other LDH (***13***). Comparative models were constructed for TvLDH and TvMDH to study the sequences in a structural context and to suggest site-directed mutagenesis experiments to elucidate changes in enzymatic specificity in this apparent case of convergent evolution. The native and mutated enzymes were subsequently expressed and their activities compared (***13***).

3.1. *Fold Assignment*

1. The first step in comparative modeling is to identify one or more template structure(s) that have detectable similarity to the target. This identification is achieved by scanning the sequence of TvLDH against a library of sequences extracted from known protein structures in the Protein Data Bank (PDB; (***14***)). This step is performed using the `profile.build()` module of MODELLER (file `build_profile.py` & *see* **Note 2**). The `profile.build()` command uses the local dynamic programming algorithm to identify related sequences (***6, 15***). In the simplest case, `profile.build()` takes as input the target sequence (file `TvLDH.pir`) and a database of sequences of known structure (file `pdb_95.pir`) and returns a set of statistically significant alignments (file `build_profile.prf`). Execute the command by typing `mod8v2 build_profile.py`.

2. The results of the scan are stored in the output file called `build_profile.prf`. The first six lines of this file contain the input parameters used to create the alignments. Subsequent lines contain several columns of data; for the purposes of this example, the most important columns are (i) the second column, containing the PDB code of the related template sequences; (ii) the eleventh column, containing the percentage sequence identity between the TvLDH and template sequences; and (iii) the twelfth column, containing the E-values for the statistical significance of the alignments.

3. The extent of similarity between the target-template pairs is usually quantified using sequence identity or a statistical measure such as E-value (*see* **Notes 3, 4**). Inspection of column 11 shows that the template with the highest sequence identity with the target is the 1y7tA structure (45% sequence identity). Further inspection of column 12 shows that there are six PDB sequences, all corresponding to malate dehydrogenases (1y7tA, 5mdhA, 1b8pA, 1civA, 7mdhA and 1smkA) that show significant similarities to TvLDH with E-values of zero. Two variations of the model building procedure will be described below, one using a single template with the highest sequence identity (1y7tA), and another using all six templates (3.2.2), to highlight their differences (*see* **Note 5**).

3.2. *Sequence-Structure Alignment*

Sequence-structure alignments will be calculated using the `align2d()` module of MODELLER (*see* **Note 6**). Although `align2d()` is based on a global dynamic programming algorithm (***16***), it is different from standard sequence-sequence alignment methods because it takes into account structural information from the template when constructing an alignment. This task is achieved through a variable gap penalty function that tends to place gaps in solvent exposed and curved regions, outside secondary structure segments, and between two positions that are close in space (***9***). In the current example, the target-template similarity is so high that almost any method with reasonable parameters will result in the correct alignment (*see* **Note 7**).

3.2.1. *Single-Template*

1. The input script `align2d-single.py` reads in the structure of the chosen template (1y7tA) and the target sequence (TvLDH) and calls the `align2d()` module to perform the alignment. The resulting alignment is written out to the specified alignment files (`TvLDH-1y7tA.ali` in the PIR format and `TvLDH-1y7tA.pap` in the PAP format).

3.2.2. *Multiple-Template*

1. The first step in using multiple templates for modeling is to obtain a multiple structure alignment of all the chosen templates. The structure alignment module of MODELLER, `salign()`, can be used for this purpose (***17***). The input script `salign.py` contains the necessary Python instructions to achieve a multiple structure alignment. The script reads in all the six template structures into an alignment object and then calls `salign()` to generate the multiple structure alignment. The output alignment is written out to `TvLDH-salign.ali` and `TvLDH-salign.pap`, in the PIR and PAP formats, respectively.

2. The next step is to align the TvLDH sequence with the multiple structure alignment generated above. This task is accomplished using the script file `align2d-multiple.py,` that again calls the `align2d()` module to calculate the sequence-structure alignment. Upon execution, the resulting alignments are written to `TvLDH-multiple.ali` and `TvLDH-multiple.pap` in the PIR and PAP formats, respectively.

3.3. *Model Building*

Two variations of the model building protocol will be described, corresponding to the two alignments generated above: (i) modeling using a single template and (ii) modeling using multiple templates, followed by building and optimizing a consensus model. The files required for each of these protocols are present in separate subdirectories called `single/` and `multiple/`, respectively.

3.3.1. *Single Template*

1. The input script `model-single.py` lists the Python commands necessary to build the model of the TvLDH sequence using the information derived from 1y7tA structure. The script calls the `automodel` class specifying the name of the alignment file to use and the identifiers of the target (TvLDH) and template sequences (1y7tA). The `starting_model` and `ending_model` specify that ten models should be calculated by randomizing the initial coordinates. The models

are then assessed with the GA341 (***18, 19***) and DOPE assessment functions (***12***).

2. Upon completion, the 10 models for the TvLDH are written out in the PDB format to files called `TvLDH.B9990[0001-0010].pdb` (*see* **Notes 8, 9**).

3.3.2. *Multiple Templates with Consensus Modeling*

1. The input script, `model-multiple.py`, is quite similar to `model-single.py`. The specification of the template codes to `automodel` now contains all six chosen PDB codes and additionally, the `cluster()` method is called to exploit the diversity of the 10 generated models *via* a clustering and optimization procedure to construct a single consensus model (*see* **Note 10**).

2. Upon completion, the 10 models for the TvLDH and the consensus model are written out to `TvLDH.B9990[0001-0010].pdb` and `cluster.opt`, respectively.

3.4. *Model Evaluation*

1. The log files produced by each of the model building procedures (`model-single.log` and `model-multiple.log`) contain a summary of each calculation at the bottom of the file. This summary includes, for each of the 10 models, the MODELLER objective function (*see* **Note 11**) (***5***), the DOPE pseudo-energy value (*see* **Note 12**), and the value of the GA341 score (*see* **Notes 13, 14**). These scores can be used to identify which of the 10 models produced is likely to be the most accurate model (*see* **Note 15**).

2. A residue-based pseudo-energy profile for the best scoring model, chosen as the one with the lowest DOPE statistical potential score, can be obtained by executing the `evaluate_model.py` script. This script is available in each of the subdirectories mentioned above. Such a profile is useful to detect local regions of high pseudo-energy that usually correspond to errors in the model (*see* **Notes 16, 17**).

Fig. 2 shows the pseudo-energy profiles of the best scoring models from each procedure. It can be seen that some of the errors in the single-template model have been resolved in the model calculated using multiple templates.

4. Notes

1. Exactly the same job run on two different types of computers (*eg*, Windows/Intel and a Macintosh) generally returns slightly different results. The reason for this variation is the difference in the rounding of floating point numbers, which may lead to a divergence between optimization trajectories starting at exactly the same initial conditions. Though these differences are generally small, for absolute reproducibility, the same type of computer architecture and operating system need to be used.

2. As mentioned earlier, knowledge of the Python scripting language is not a requirement for basic use of MODELLER. The lines in the script file are usually self-explanatory and input/output options for each module are described in the manual. For the purpose of illustration, the various lines of the `build_profile.py` script are described below (**Fig. 3**):

   - `log.verbose()` sets the amount of information that is written out to the log file.

   - `environ()` initializes the 'environment' for the current modeling run, by creating a new environ object, called `env`. Almost all MODELLER scripts require this step, as the new object is needed to build most other useful objects.

   - `sequence_db()` creates a sequence database object, calling it `sdb`, which is used to contain large databases of protein sequences.

   - `sdb.read()` reads a file, in text format, containing non-redundant PDB sequences into the `sdb` database. The input options to this command specify the name of the file (`seq_database_file='pdb_95.pir'`), the format of the file (`seq_database_format='pir'`), whether to read all sequences from the file (`chains_list='all'`), upper and lower bounds for the lengths of the sequences to be read (`minmax_db_seq_len=(30,3000)`), and whether to clean the sequences of non-standard residue names (`clean_sequences=True`).

- `sdb.write()` writes a binary machine-independent file (`seq_database_format='binary'`) with the specified name (`seq_database_file='pdb_95.bin'`), containing all sequences read in the previous step.

- The second call to `sdb.read()` reads the binary format file back in for faster execution.

- `alignment()` creates a new 'alignment' object (`aln`).

- `aln.append()` reads the target sequence TvLDH from the file `TvLDH.ali` and `aln.to_profile()` converts it to a profile object (`prf`). Profiles contain similar information as alignments, but are more compact and better suited for sequence database searching.

- `prf.build()` searches the sequence database (`sdb`) using the target profile stored in the `prf` object as the query. Several options, such as the parameters for the alignment algorithm (`matrix_offset`, `rr_file`, `gap_penalties` *etc*.), are specified to override the default settings. `max_aln_evalue` specifies the threshold value to use when reporting statistically significant alignments.

- `prf.write()` writes a new profile containing the target sequence and its homologs into the specified output file (`file=build_profile.prf`).

- The profile is converted back to the standard alignment format and written out using `aln.write()`.

3. Sequence-structure relationships can be divided into three different regimes of the sequence similarity spectrum: (i) the easily detected relationships character-

ized by >30% sequence identity, (ii) the "twilight zone" (***20***) corresponding to relationships with statistically significant sequence similarity, with identities generally in the 10-30% range, and (iii) the "midnight zone" (***20***) corresponding to statistically insignificant sequence similarity. Hence, the sequence identity is a good predictor of the accuracy of the final model when its value is >30%. It has been shown that models based on such alignments usually have, on average, more than ~60% of the backbone atoms correctly modeled with a root-mean-squared-deviation (RMSD) of less than 3.5 Å (**Fig. 4**).

However, the sequence identity is not a statistically reliable measure of alignment significance and corresponding model accuracy for values lower than 30% (***20, 21***). During a scan of a large database, for instance, it is possible that low values occur purely by chance. In such cases, it is useful to quantify the sequence-structure relationship using more robust measures of statistical significance, such as E-values, that compare the score obtained for an alignment with an established background distribution of such scores.

4. One other problem of using sequence identity as a measure to select templates is that, in practice, there is no single generally used way to normalize it (***21***). For instance, local alignment methods usually normalize the number of identically aligned residues by the length of the alignment, while global alignment methods normalize it by either the length of the target sequence or the length of the shorter of the two sequences. Therefore, it is possible that alignments of short fragments produce a high sequence identity but do not result in an accurate model. Measures of statistical significance do not suffer from this normalization problem because the alignment scores are always corrected for the length of the aligned segment before the significance is computed (***22, 23***).

5. After a list of all related protein structures and their alignments with the target sequence has been obtained, template structures are usually prioritized depending on the purpose of the comparative model. Template structures may be chosen

based purely on the target-template sequence identity or a combination of several other criteria, such as the experimental accuracy of the structures (resolution of x-ray structures, number of restraints per residue for NMR structures), conservation of active-site residues, holo-structures that have bound ligands of interest, and prior biological information that pertains to the solvent, *p*H, and quaternary contacts.

6. Although fold assignment and sequence-structure alignment are logically two distinct steps in the process of comparative modeling, in practice almost all fold assignment methods also provide sequence-structure alignments. In the past, fold assignment methods were optimized for better sensitivity in detecting remotely related homologs, often at the cost of alignment accuracy. However, recent methods simultaneously optimize both the sensitivity and alignment accuracy. For the sake of clarity, however, they are still considered as separate steps in the current chapter.

7. Most alignment methods use either the local or global dynamic programming algorithms to derive the optimal alignment between two or more sequences and/or structures. The methods, however, vary in terms of the scoring function that is being optimized. The differences are usually in the form of the gap-penalty function (linear, affine, or variable) (***9***), the substitution matrix used to score the aligned residues (20x20 matrices derived from alignments with a given sequence identity, those derived from structural alignments, those incorporating the structural environment of the residues) (***24***), or combinations of both (***25-28***). There doesn't yet exist a single universal scoring function that guarantees the most accurate alignment for all situations. Above 30-40% sequence identity, alignments produced by almost all of the methods are similar. However, in the twilight and midnight zones of sequence identity, models based on the alignments of different methods tend to have significant variations in accuracy. Improving the performance and accuracy of methods in this regime remains one of the main tasks of comparative modeling (***29, 30***).

The single source of errors with the largest impact on comparative modeling is misalignments, especially when the target-template sequence identity decreases below 30%. It is imperative to calculate an accurate alignment between the target-template pair, as comparative modeling can almost never recover from an alignment error (***31***).

8. Comparative models do not reflect the fluctuations of a protein structure in solution. That is, the variability seen in the structures of multiple models built for one set of inputs reflect different solutions to the molecular objective function, which do not correspond to the actual dynamics of the protein structure in nature.

9. If there are no large differences among the template structures (> 2 Å backbone RMSD) and no long insertions or deletions (> 5 residues) between the target and the template(s), building multiple models generally does not drastically improve the accuracy of the best model produced. For alignments to similar templates that lack many gapped regions, building multiple models from the same input alignment most often results in a narrow distribution of accuracies: the difference between the C^{α} RMSD values between each model and the true native structure is usually within a range of 0.5 Å for a sequence containing ~150 residues (***5***). If, however, the sequence-structure alignment contains different templates with many insertions and/or deletions, it is important to calculate multiple models for the same alignment. Calculating multiple models allows for better sampling of the different templates segments and the conformations of the unaligned regions, and will often result in a more accurate model than if only one model had been produced.

10. A consensus model is calculated by first clustering an ensemble of models and then averaging individual atomic positions. The consensus model is then optimized using the same protocol used on the individual models. Construction of a consensus model followed by optimization usually results in a model with a lower

objective function than any of the contributing models; the construction of a consensus model can thus be seen as a part of an efficient optimization. When there are substantial variations in regions of the contributing models, due to the variation among the templates and the presence of gaps in the alignment, calculating the consensus using cluster averaging usually produces the most accurate conformation.

11. The MODELLER objective function is a measure of how well the model satisfies the input spatial restraints. Lower values of the objective function indicate a better fit with the input data and, thus, models that are likely to be more accurate (***5***).

12. The Discrete Optimized Protein Energy (DOPE) (***12***) is an atomic distance-dependant statistical potential based on a physical reference state that accounts for the finite size and spherical shape of proteins. The reference state assumes a protein chain consists of non-interacting atoms in a homogeneous sphere of equivalent radius to that of the corresponding protein. The DOPE potential was derived by comparing the distance statistics from a non-redundant PDB subset of 1,472 high-resolution protein structures with the distance distribution function of the reference state. By default, the DOPE score is not included in the model building routine, and thus can be used as an independent assessment of the accuracy of the output models. The DOPE score assigns a score for a model by considering the positions of all non-hydrogen atoms, with lower scores corresponding to models that are predicted to be more accurate.

13. The GA341 criterion is a composite fold-assessment score that combines a Z-score calculated with a statistical potential function, target-template sequence identity, and a measure of structural compactness (***18, 19***). The score ranges from 0.0 for models that tend to have an incorrect fold to 1.0 for models that tend to be comparable to low-resolution x-ray structures. Comparison of models with their corresponding experimental structures indicates that models with GA341 scores greater than 0.7 generally have the correct fold with more than 35% of the

backbone atoms superposable within 3.5Å of their native positions. Reliable models (GA341 score ≥ 0.7) based on alignments with more than 40% sequence identity, have a median overlap of more than 90% with the corresponding experimental structure. In the 30-40% sequence identity range, the overlap is usually between 75-90% and below 30% it drops to 50-75%, or even less in the worst cases.

14. The accuracy of a model should first be assessed using the GA341 score to increase or decrease our confidence in the fold of the model. An assessment of an incorrect fold implies that an incorrect template(s) was chosen or an incorrect alignment with the correct template was used for model calculation. When the target-template relationship falls in the twilight or midnight zones, it is usually difficult to differentiate between these two kinds of errors. In such cases, building models based on different sets of templates may resolve the problem.

15. Different measures to predict errors in a protein structure perform best at different levels of resolution. For instance, physics-based force-fields may be helpful at identifying the best model when all models are very close to the native state (< 1.5 Å RMSD, corresponding to ~85% target-template sequence identity). In contrast, coarse-grained scores such as distance-based statistical potentials have been shown to have the greatest ability to differentiate between models in the ~3 Å C^{α} RMSD range. Tests show that such scores are often able to identify a model within 0.5 Å C^{α} RMSD of the most accurate model produced (***32***). When multiple models are built, the DOPE score generally selects a more accurate model than the MODELLER objective function.

16. Segments of the target sequence that have no equivalent region in the template structure (*ie*, insertions or loops) are among the most difficult regions to model (***4, 33-35***). This difficulty is compounded when the target and template are distantly related, with errors in the alignment leading to incorrect positions of the insertions and distortions in the loop environment. Using alignment methods that incorpo-

rate structural information can often correct such errors (***9***). Once a reliable alignment is obtained, various modeling protocols can predict the loop conformation, for insertions of less than 8-10 residues long (***4, 33, 36, 37***).

17. As a consequence of sequence divergence, the mainchain conformation of a protein can change, even if the overall fold remains the same. Therefore, it is possible that in some correctly aligned segments of a model, the template is locally different (< 3 Å) from the target, resulting in errors in that region. The structural differences are sometimes not due to differences in sequence, but are a consequence of artifacts in structure determination or structure determination in different environments (*eg*, packing of subunits in a crystal). The simultaneous use of several templates can minimize this kind of an error (***38, 39***).

5. References

1. Baker, D., and Sali, A. (2001) Protein structure prediction and structural genomics. *Science* **294**, 93-96.
2. Pieper, U., Eswar, N., Davis, F. P., Braberg, H., Madhusudhan, M. S., Rossi, A., Marti-Renom, M., Karchin, R., Webb, B. M., Eramian, D., Shen, M. Y., Kelly, L., Melo, F., and Sali, A. (2006) MODBASE: a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res.* **34**, D291-295.
3. Marti-Renom, M. A., Stuart, A. C., Fiser, A., Sanchez, R., Melo, F., and Sali, A. (2000) Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.* **29**, 291-325.
4. Fiser, A., Do, R. K., and Sali, A. (2000) Modeling of loops in protein structures. *Protein Sci.* **9**, 1753-1773.
5. Sali, A., and Blundell, T. L. (1993) Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **234**, 779-815.
6. Eswar, N., Madhusudhan, M.S., Marti-Renom, M.A., Sali, A. (2005) BUILD_PROFILE: A module for calculating sequence profiles in MODELLER. *http://www.salilab.org/modeller.*
7. Marti-Renom, M. A., Madhusudhan, M. S., and Sali, A. (2004) Alignment of protein sequences by their profiles. *Protein Sci.* **13**, 1071-1087.
8. Eswar, N., Madhusudhan, M.S., Marti-Renom, M.A., Sali, A. (2005) PROFILE_SCAN: A module for fold-assignment using profile-profile scanning in MODELLER. *http://www.salilab.org/modeller.*
9. Madhusudhan, M. S., Marti-Renom, M. A., Sanchez, R., and Sali, A. (2006) Variable gap penalty for protein sequence-structure alignment. *Protein Eng. Des. Sel.* **19**, 129-133.

10. MacKerell, A. D., Jr., Bashford, D., Bellott, M., Dunbrack, R. L., Jr., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Muczera, K., Lau, F. T. K., Mattos, C., Michnik, S., Nguyen, D. T., Ngo, T., Prodhom, B., reiher, W. E., III, Roux, B., Schlenkrich, M., Smith, J. C., Stote, R., Straub, J., Watanabe, M., Wiorkiewicz-Kuczera, J., Yin, D., and Karplus, M. (1998) All-atom empirical potential for molecular modleing and dynamics studies of proteins. *J.Phys.Chem.B.* **102**, 3586-3616.
11. Sali, A., and Overington, J. P. (1994) Derivation of rules for comparative protein modeling from a database of protein structure alignments. *Protein Sci* 3, 1582-1596.
12. Shen, M. Y., and Sali, A. (2006) Statistical potential for assessment and prediction of protein structures. *Protein Sci.* **15**, 2507-2524.
13. Wu, G., Fiser, A., ter Kuile, B., Sali, A., and Muller, M. (1999) Convergent evolution of Trichomonas vaginalis lactate dehydrogenase from malate dehydrogenase. *Proc. Natl. Acad. Sci. U. S. A.* **96**, 6285-6290.
14. Deshpande, N., Addess, K. J., Bluhm, W. F., Merino-Ott, J. C., Townsend-Merino, W., Zhang, Q., Knezevich, C., Xie, L., Chen, L., Feng, Z., Green, R. K., Flippen-Anderson, J. L., Westbrook, J., Berman, H. M., and Bourne, P. E. (2005) The RCSB Protein Data Bank: a redesigned query system and relational database based on the mmCIF schema. *Nucleic Acids Res.* **33**, D233-237.
15. Smith, T. F., and Waterman, M. S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195-197.
16. Needleman, S. B., and Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443-453.
17. Madhusudhan, M. S., Eswar, N., Marti-Renom, M.A., Sali, A. (2005) SALIGN: A module for multiple sequence/structure alignments in MODELLER. *http://www.salilab.org/modeller.*
18. John, B., and Sali, A. (2003) Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic Acids Res* **31**, 3982-3992.
19. Melo, F., Sanchez, R., and Sali, A. (2002) Statistical potentials for fold assessment. *Protein Sci.* **11**, 430-448.
20. Rost, B. (1999) Twilight zone of protein sequence alignments. *Protein Eng.* **12**, 85-94.
21. May, A. C. (2004) Percent sequence identity; the need to be explicit. *Structure* **12**, 737-738.
22. Altschul, S. F., Madden, T. L., Schaffer, 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.
23. Pearson, W. R. (1998) Empirical statistical estimates for sequence similarity searches. *J. Mol. Biol.* **276**, 71-84.
24. Henikoff, S., and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.* **89**, 10915-10919.

25. Zhou, H., and Zhou, Y. (2005) Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins* **58**, 321-328.
26. McGuffin, L. J., and Jones, D. T. (2003) Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics* **19**, 874-881.
27. Karchin, R., Cline, M., Mandel-Gutfreund, Y., and Karplus, K. (2003) Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins* **51**, 504-514.
28. Shi, J., Blundell, T. L., and Mizuguchi, K. (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.* **310**, 243-257.
29. Dunbrack, R. L., Jr. (2006) Sequence comparison and protein structure prediction. *Curr. Opin. Struct. Biol.* **16**, 374-384.
30. Xiang, Z. (2006) Advances in homology protein structure modeling. *Curr. Protein Pept. Sci.* **7**, 217-227.
31. Sanchez, R., and Sali, A. (1997) Advances in comparative protein-structure modelling. *Curr. Opin. Struct. Biol.* **7**, 206-214.
32. Eramian, D., Shen, M. Y., Devos, D., Melo, F., Sali, A., and Marti-Renom, M. A. (2006) A composite score for predicting errors in protein structure models. *Protein Sci.* **15**, 1653-1666.
33. Jacobson, M. P., Pincus, D. L., Rapp, C. S., Day, T. J., Honig, B., Shaw, D. E., and Friesner, R. A. (2004) A hierarchical approach to all-atom protein loop prediction. *Proteins* **55**, 351-367.
34. Fernandez-Fuentes, N., Oliva, B., and Fiser, A. (2006) A supersecondary structure library and search algorithm for modeling loops in protein structures. *Nucleic Acids Res.* **34**, 2085-2097.
35. Zhu, K., Pincus, D. L., Zhao, S., and Friesner, R. A. (2006) Long loop prediction using the protein local optimization program. *Proteins* **65**, 438-452.
36. Coutsias, E. A., Seok, C., Jacobson, M. P., and Dill, K. A. (2004) A kinematic view of loop closure. *J. Comput. Chem.* **25**, 510-528.
37. van Vlijmen, H. W., and Karplus, M. (1997) PDB-based protein loop prediction: parameters for selection and methods for optimization. *J. Mol. Biol.* **267**, 975-1001.
38. Sanchez, R., Sali, A. (1997) Evaluation of comparative protein structure modeling by MODELLER-3. *Proteins* Suppl **1**, 50-58.
39. Srinivasan, N., and Blundell, T. L. (1993) An evaluation of the performance of an automated procedure for comparative modelling of protein tertiary structure. *Protein Eng.* **6**, 501-512.
40. Sanchez, R., Sali, A. (1998) Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. *Proc. Natl. Acad. Sci. U. S. A.* **95**, 13597-13602.
41. Chothia, C., and Lesk, A. M. (1986) The relation between the divergence of sequence and structure in proteins. *Embo. J.* **5**, 823-826.

Legends to Figures:

Fig. 1. Comparative protein structure modeling. (a) A flowchart illustrating the steps in the construction of a comparative model (***3***). (b) Description of comparative modeling by extraction of spatial restraints as implemented in MODELLER (***5***). By default, spatial restraints in MODELLER involve (i) homology-derived restraints from the aligned template structures, (ii) statistical restraints derived from all known protein structures, and (iii) stereochemical restraints from the CHARMM-22 molecular mechanics force field. These restraints are combined into an objective function that is then optimized to calculate the final 3D structure of the target sequence.

Fig. 2. The four steps of comparative modeling as applied to the described example. (a) Scanning the sequence of TvLDH against the sequences of PDB structures identifies 1y7tA as a single template with highest sequence identity; six other malate dehydrogenases are also identified with statistically significant E-values; (b) sequence-sequence structure alignments are generated using a variable gap penalty method; (c) ten models are constructed per alignment and the best model is chosen using the DOPE statistical potential; in addition, a consensus model is calculated from the ten models constructed using multiple templates. The model based on single template is shown in black, the one based on multiple templates is in dark-grey and the consensus model is shown in light-grey; (d) the residue-averaged DOPE scores are used to evaluate local regions of high, unfavorable score that tend to correspond to errors. The consensus model results in the best profile (black line).

Fig. 3. The Python input script used for fold assignment. **See Note 2** for details.

Fig. 4. Average model accuracy as a function of sequence identity (***40***). As the sequence identity between the target sequence and the template structure decreases, the average structural similarity between the template and the target also decreases (dark grey area, squares) (***41***). Structural overlap is defined as the fraction of equivalent C^{α} atoms. For the comparison of the model with the actual structure (circles), two C^{α} atoms were considered equivalent if they belonged to the same residue and were within 3.5 Å of each other after least squares superposition. For comparisons between the template structure and the actual target structure (squares), two C^{α} atoms were considered equivalent if they were within 3.5 Å of each other after alignment and rigid-body superposition. The difference between the model and the actual target structure is a combination of the target-template differences (dark grey area) and the alignment errors (light grey area). The figure was constructed by calculating ~1 million comparative models based on single template of varying similarity to the targets. All targets had known (experimentally determined) structures.

Figure 1

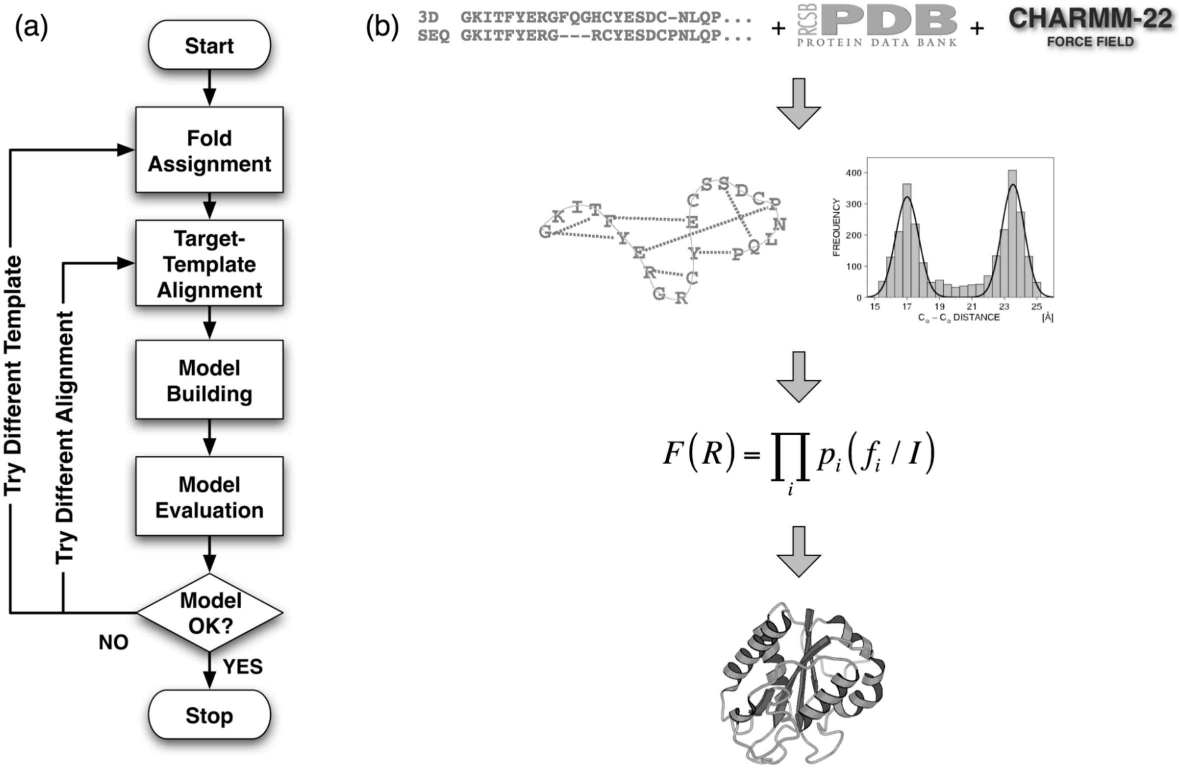
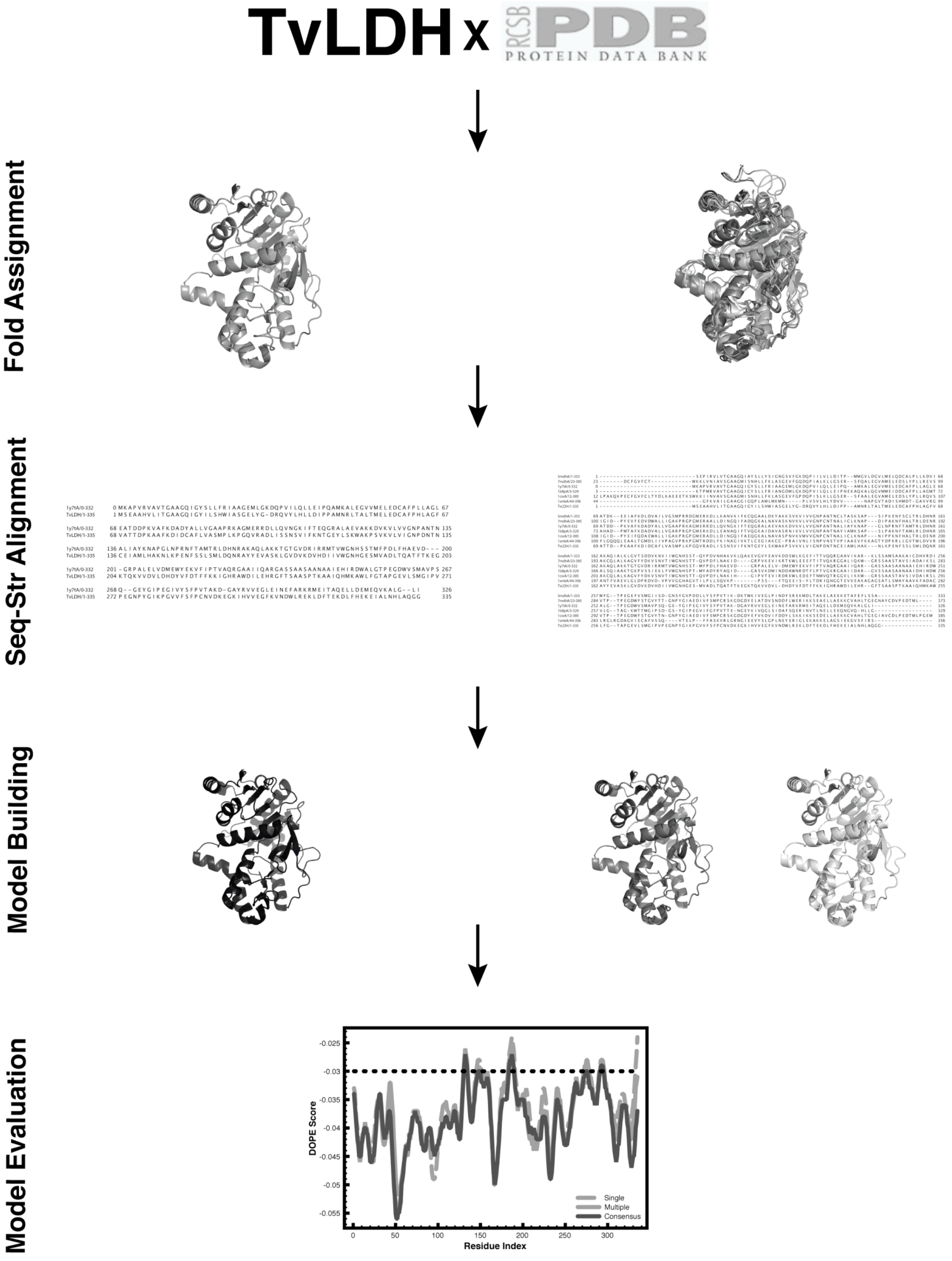

(a)
Start
Fold Assignment
Target-Template Alignment
Model Building
Model Evaluation
Model OK?
NO
YES
Stop
Try Different Template
Try Different Alignment
(b)
3D GKITFYERGFQGHCYESDC-NLQP...
SEQ GKITFYERG---RCYESDCPNLQP...
+
RCSB PDB
PROTEIN DATA BANK
+
CHARMM-22
FORCE FIELD
400
300
200
100
0
FREQUENCY
15
17
19
21
23
25
$C_\alpha - C_\alpha$ DISTANCE
[Å]
$F(R) = \prod_i p_i(f_i / I)$

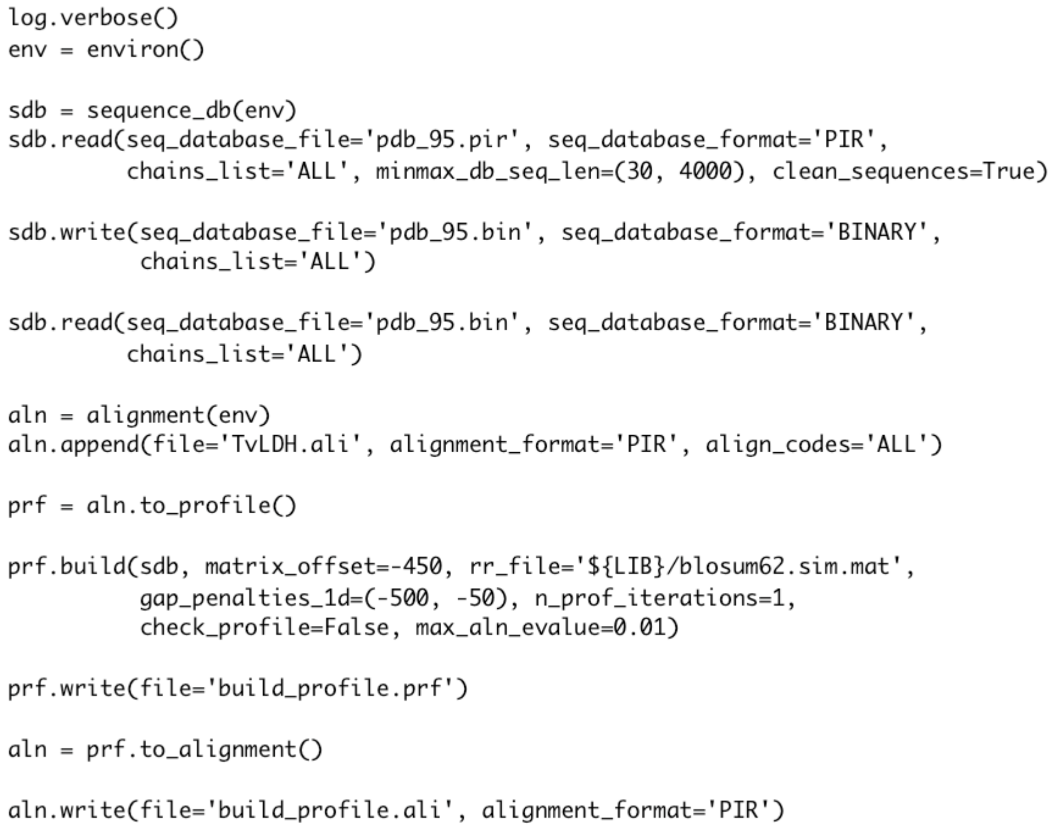
Figure 2



```
1y7tA/0-332    0 MKAPVRVAVTGAAGQIGYSLLFRIAAGEMLGKDQPVILQLLEIPQAMKALEGVVMELEDCAFPLLAGL 67
TvLDH/1-335    1 MSEAAHVLITGAAGQIGYILSHWIASGELYG-DRQVYLHLLDIPPAMNRLTALTMELEDCAFPHLAGF 67

1y7tA/0-332   68 EATDDPKVAFKDADYALLVGAAPRKAGMERRDLLQVNGKIFTEQGRALAEVAKKDVKVLVVGNPANTN 135
TvLDH/1-335   68 VATTDPKAAFKDIDCAFLVASMPLKPGQVRADLISSNSVIFKNTGEYLSKWAKPSVKVLVIGNPDNTN 135

1y7tA/0-332  136 ALIAYKNAPGLNPRNFTAMTRLDHNRAKAQLAKKTGTGVDRIRRMTVWGNHSSTMFPDLFHAEVD--- 200
TvLDH/1-335  136 CEIAMLHAKNLKPENFSSLSMLDQNRAYYEVASKLGVDVKDVHDIIVWGNHGESMVADLTQATFTKEG 203

1y7tA/0-332  201 -GRPALELVDMEWYEKVFIPTVAQRGAAIIQARGASSAASAANAAIEHIRDWALGTPEGDWVSMAVPS 267
TvLDH/1-335  204 KTQKVVDVLDHDYVFDTFFKKIGHRAWDILEHRGFTSAASPTKAAIQHMKAWLFGTAPGEVLSMGIPV 271

1y7tA/0-332  268 Q--GEYGIPEGIVYSFPVTAKD-GAYRVVEGLEINEFARKRMEITAQELLDEMEQVKALG--LI     326
TvLDH/1-335  272 PEGNPYGIKPGVVFSFPCNVDKEGKIHVVEGFKVNDWLREKLDFTEKDLFHEKEIALNHLAQGG     335
```

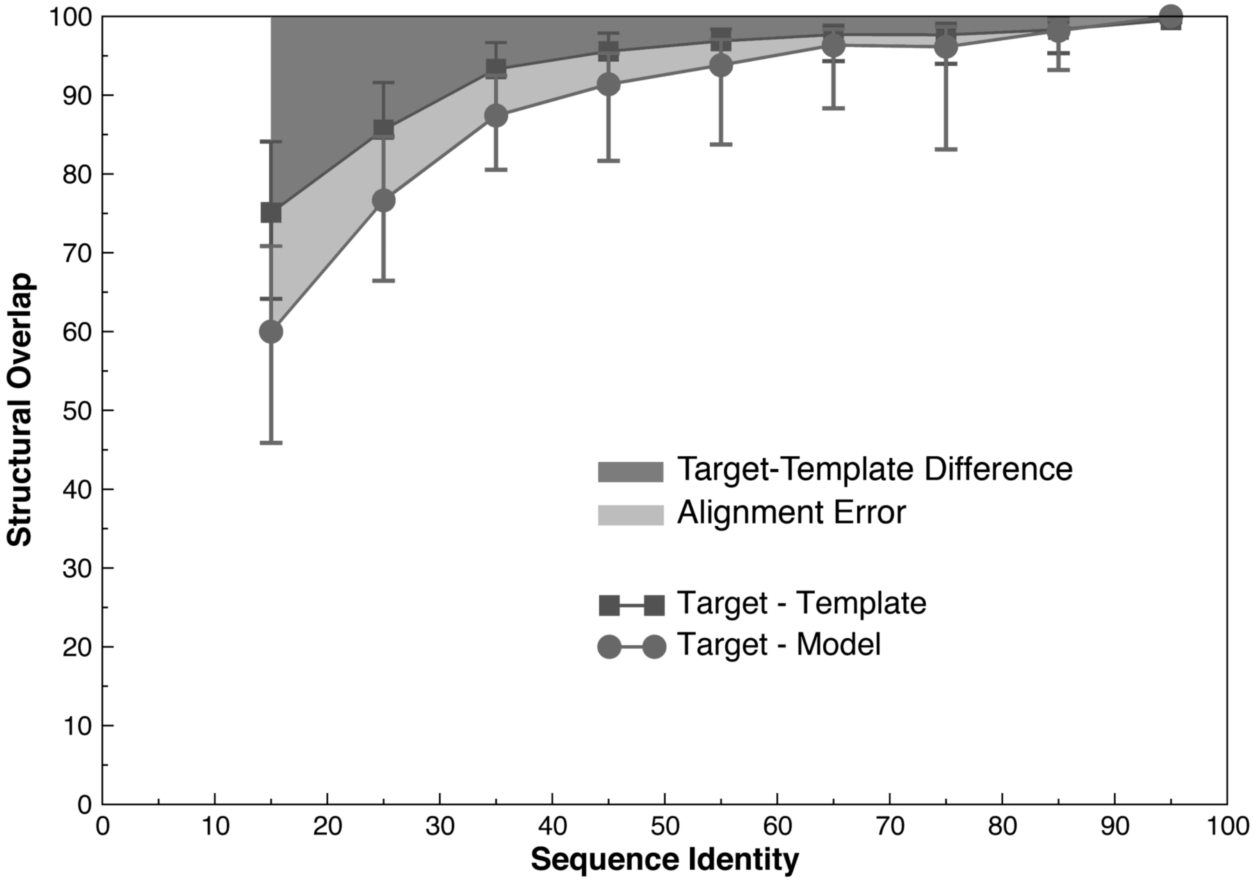
```
5mdhA/1-333    1 ----------------------------SEPIRVLVTGAAGQIAYSLLYSIGNGSVFGKDQPIILVLLDITP--MMGVLDGVLMELQDCALPLLKDVI 68
7mdhA/23-385  23 -------DCFGVFCT------------WKKLVNIAVSGAAGMISNHLLFKLASGEVFGQDQPIALKLLGSER--SFQALEGVAMELEDSLYPLLREVS 99
1y7tA/0-332    0 ---------------------------MKAPVRVAVTGAAGQIGYSLLFRIAAGEMLGKDQPVILQLLEIPQ--AMKALEGVVMELEDCAFPLLAGLE 68
1b8pA/3-329    3 ----------------------------KTPMRVAVTGAAGQICYSLLFRIANGDMLGKDQPVILQLLEIPNEKAQKALQGVMMEIDDCAFPLLAGMT 72
1civA/12-385  12 LPAKQKPECFGVFCLTYDLKAEEETKSWKKIINVAVSGAAGMISNHLLFKLASGEVFGPDQPISLKLLGSER--SFAALEGVAMELEDSLYPLLRQVS 107
1smkA/44-356  44 -------------------------------GFKVAILGAAGGIGQPLAMLMKMN-----PLVSVLHLYDVV------NAPGVTADISHMDT-GAVVRG 99
TvLDH/1-335    1 ---------------------------MSEAAHVLITGAAGQIGYILSHWIASGELYG-DRQVYLHLLDIPP--AMNRLTALTMELEDCAFPHLAGFV 68

5mdhA/1-333   69 ATDK--EEIAFKDLDVAILVGSMPRRDGMERKDLLKANVKIFKCQGAALDKYAKKSVKVIVVGNPANTNCLTASKSAP---SIPKENFSCLTRLDHNR 161
7mdhA/23-385 100 IGID--PYEVFEDVDWALLIGAKPRGPGMERAALLDINGQIFADQGKALNAVASKNVKVLVVGNPCNTNALICLKNAP---DIPAKNFHALTRLDENR 192
1y7tA/0-332   69 ATDD--PKVAFKDADYALLVGAAPRKAGMERRDLLQVNGKIFTEQGRALAEVAKKDVKVLVVGNPANTNALIAYKNAP---GLNPRNFTAMTRLDHNR 161
1b8pA/3-329   73 AHAD--PMTAFKDADVALLVGARPRGPGMERKDLLEANAQIFTVQGKAIDAVASRNIKVLVVGNPANTNAYIAMKSAP---SLPAKNFTAMLRLDHNR 165
1civA/12-385 108 IGID--PYEIFQDAEWALLIGAKPRGPGMERADLLDINGQIFAEQGKALNAVASPNVKVMVVGNPCNTNALICLKNAP---NIPPKNFHALTRLDENR 200
1smkA/44-356 100 FLGQQQLEAALTGMDLIIVPAGVPRKPGMTRDDLFKINAGIVKTLCEGIAKCC-PRAIVNLISNPVNSTVPIAAEVFKKAGTYDPKRLLGVTMLDVVR 196
TvLDH/1-335   69 ATTD--PKAAFKDIDCAFLVASMPLKPGQVRADLISSNSVIFKNTGEYLSKWAKPSVKVLVIGNPDNTNCEIAMLHAK---NLKPENFSSLSMLDQNR 161

5mdhA/1-333  162 AKAQIALKLGVTSDDVKNVIIWGNHSST-QYPDVNHAKVKLQAKEVGVYEAVKDDSWLKGEFITTVQQRGAAVIKAR--KLSSAMSAAKAICDHVRDI 256
7mdhA/23-385 193 AKCQLALKAGVFYDKVSNVTIWGNHSTT-QVPDFLNAKID----GRPVKEVIKRTKWLEEEFTITVQKRGGALIQKW--GRSSAASTAVSIADAIKSL 283
1y7tA/0-332  162 AKAQLAKKTGTGVDRIRRMTVWGNHSST-MFPDLFHAEVD----GRPALELV-DMEWYEKVFIPTVAQRGAAIIQAR--GASSAASAANAAIEHIRDW 251
1b8pA/3-329  166 ALSQIAAKTGKPVSSIEKLFVWGNHSPT-MYADYRYAQID----GASVKDMINDDAWNRDTFLPTVGKRGAAIIDAR--GVSSAASAANAAIDHIHDW 256
1civA/12-385 201 AKCQLALKAGVFYDKVSNVTIWGNHSTT-QVPDFLNAKIH----GIPVTEVIRDRKWLEDEFTNMVQTRGGVLIKKW--GRSSAASTAVSIVDAIRSL 291
1smkA/44-356 197 ANTFVAEVLGLDPRDVD-VPVVGGHAGVTILPLLSQVKP-------PSS---FTQEEIS-YLTDRIQNGGTEVVEAKAGAGSATLSMAYAAVKFADAC 282
TvLDH/1-335  162 AYYEVASKLGVDVKDVHDIIVWGNHGES-MVADLTQATFTKEGKTQKVVDVL-DHDYVFDTFFKKIGHRAWDILEHR--GFTSAASPTKAAIQHMKAW 255

5mdhA/1-333  257 WFG--TPEGEFVSMGIISD-GNSYGVPDDLLYSFPVTIK-DKTWKIVEGLPINDFSREKMDLTAKELAEEKETAFEFLSSA-----------------  333
7mdhA/23-385 284 VTP--TPEGDWFSTGVYTT-GNPYGIAEDIVFSMPCRSKGDGDYELATDVSNDDFLWERIKKSEAELLAEKKCVAHLTGEGNAYCDVPEDTML----  373
1y7tA/0-332  252 ALG--TPEGDWVSMAVPSQ-GE-YGIPEGIVYSFPVTAK-DGAYRVVEGLEINEFARKRMEITAQELLDEMEQVKALGLI------------------  326
1b8pA/3-329  257 VLG--TAG-KWTTMGIPSD-GS-YGIPEGVIFGFPVTTE-NGEYKIVQGLSIDAFSQERINVTLNELLEEQNGVQ-HLLG-----------------  329
1civA/12-385 292 VTP--TPEGDWFSTGVYTN-GNPYGIAEDIVFSMPCRSKGDGDYEFVKDVIFDDYLSKKIKKSEDELLAEKKCVAHLTGEGIAVCDLPEDTMLPGEM  385
1smkA/44-356 283 LRGLRGDAGVIECAFVSSQ----VTELP--FFASKVRLGRNGIEEVYSLGPLNEYERIGLEKAKKELAGSIEKGVSFIRS------------------  356
TvLDH/1-335  256 LFG--TAPGEVLSMGIPVPEGNPYGIKPGVVFSFPCNVDKEGKIHVVEGFKVNDWLREKLDFTEKDLFHEKEIALNHLAQGG----------------  335
```

Figure 3



```
log.verbose()
env = environ()

sdb = sequence_db(env)
sdb.read(seq_database_file='pdb_95.pir', seq_database_format='PIR',
         chains_list='ALL', minmax_db_seq_len=(30, 4000), clean_sequences=True)

sdb.write(seq_database_file='pdb_95.bin', seq_database_format='BINARY',
          chains_list='ALL')

sdb.read(seq_database_file='pdb_95.bin', seq_database_format='BINARY',
         chains_list='ALL')

aln = alignment(env)
aln.append(file='TvLDH.ali', alignment_format='PIR', align_codes='ALL')

prf = aln.to_profile()

prf.build(sdb, matrix_offset=-450, rr_file='${LIB}/blosum62.sim.mat',
          gap_penalties_1d=(-500, -50), n_prof_iterations=1,
          check_profile=False, max_aln_evalue=0.01)

prf.write(file='build_profile.prf')

aln = prf.to_alignment()

aln.write(file='build_profile.ali', alignment_format='PIR')
```

Figure 4


100
90
80
70
60
50
40
30
20
10
0
Structural Overlap
0
10
20
30
40
50
60
70
80
90
100
Sequence Identity
Target-Template Difference
Alignment Error
Target - Template
Target - Model