Chapter 1

Protein Structure Modeling with *MODELLER*

Benjamin Webb and Andrej Sali

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 at atomic resolution using experimental structure determination methods. Computational protein structure modeling techniques have the potential to bridge this sequence–structure gap. In this chapter, we present an example that illustrates the use of MODELLER to construct a comparative model for a protein with unknown structure. Automation of a similar protocol 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

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, 2]. Such models are generally produced using either comparative modeling methods or free modeling techniques (also referred to as ab initio or de novo modeling) [3]. Comparative modeling relies on structural information from related proteins to guide the modeling procedure [4–6]. Free modeling does not require a related protein but instead uses a variety of methods to combine physics with the known behaviors of protein structures (for example by combining multiple short

Daisuke Kihara (ed.), *Protein Structure Prediction*, Methods in Molecular Biology, vol. 1137,
DOI 10.1007/978-1-4939-0366-5_1, © Springer Science+Business Media New York 2014

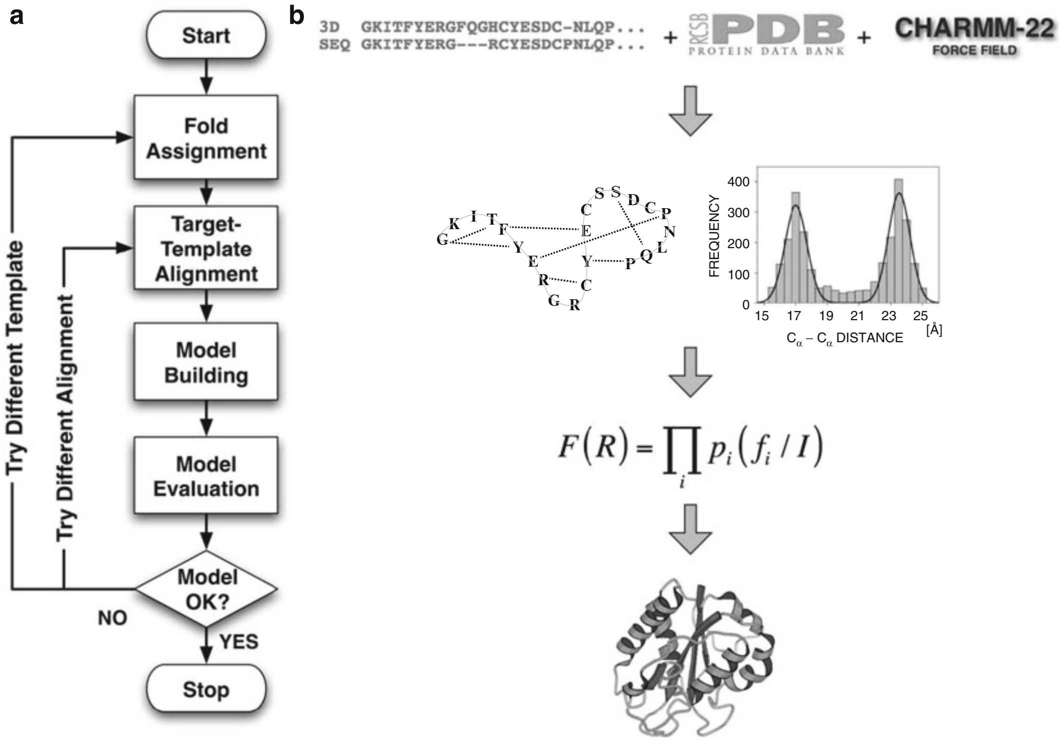

Fig. 1 Comparative protein structure modeling. (**a**) A flow chart illustrating the steps in the construction of a comparative model [4]. (**b**) Description of comparative modeling by extraction of spatial restraints as implemented in MODELLER [12]. By default, spatial restraints in MODELLER involve (1) homology-derived restraints from the aligned template structures, (2) statistical restraints derived from all known protein structures, and (3) 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 model of the target sequence

structural fragments extracted from known proteins) [7–9]; it is, however, extremely computationally expensive [3]. Comparative protein structure modeling, which this text focuses on, has been used to produce reliable structure models for at least one domain in more than half of all known sequences [10]. 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 [4] (Fig. 1): (1) fold assignment that identifies overall similarity between the target sequence and at least one known structure (template); (2) alignment of the target sequence and the template(s); (3) building a model based on the alignment with the chosen template(s); and (4) predicting the accuracy of the model.

MODELLER is a computer program for comparative protein structure modeling [11, 12]. 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, alignment of two protein sequences or their profiles [13], multiple alignment of protein sequences and/or structures [14, 15], clustering of sequences and/or structures, and ab initio modeling of loops in protein structures [11].

MODELLER implements comparative protein structure modeling by satisfaction of spatial restraints that include (1) homology-derived restraints on the distances and dihedral angles in the target sequence, extracted from its alignment with the template structures [12]; (2) stereochemical restraints such as bond length and bond angle preferences, obtained from the CHARMM-22 molecular mechanics force field [16]; (3) statistical preferences for dihedral angles and non-bonded interatomic distances, obtained from a representative set of known protein structures [17, 18]; and (4) 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.

2 Materials

To follow the examples in this discussion, both the MODELLER software and a set of suitable input files are needed. The MODELLER software is free for academic use; it can be downloaded from http://salilab.org/modeller/ and is available in binary form for most common machine types and operating systems (*see* **Note 1**). This text uses MODELLER 9.11, the most recent version at the time of writing, but the examples should also work with any newer version. The example input files can be downloaded from http://salilab.org/modeller/tutorial/MMB13.zip.

All MODELLER scripts are Python scripts. Python is pre-installed on most Linux and Mac machines; Windows users can obtain it from http://www.python.org/. It is not necessary to install Python, or to have a detailed knowledge of its use, to use MODELLER, but it is helpful for creating and understanding the more advanced MODELLER scripts.

3 Methods

The procedure for calculating a 3D 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 [19]. 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 [19].

`Monospaced text` is used below for computer file and folder/ directory names, command lines, file contents, and variable and class names.

3.1 Fold Assignment

The first step in comparative modeling is to identify one or more templates (sequences with known 3D structure) for the modeling procedure. One way to do this is to search a database of experimentally determined structures extracted from the Protein Data Bank (PDB) [20] to find sequences that have detectable similarity to the target (*see* **Note 2**). To prepare this database (*see* **Note 3**), run the following command from the command line (*see* **Note 4**):

```
python make_pdb_95.py>make_pdb_95.log
```

This generates a file called `pdb_95.bin`, which is a binary representation of the search database (*see* **Note 5**), and a log file, `make_pdb_95.log`. Next, MODELLER's `profile.build()` command is used; this uses the local dynamic programming algorithm to identify sequences related to TvLDH [21]. In the simplest case, `profile.build()` takes as input the target sequence, in file `TvLDH.ali` (*see* **Note 6**), and the binary database and returns a set of statistically significant alignments (file `build_profile.prf`) and a MODELLER log file (`build_profile.log`). Run this step by typing

```
python build_profile.py>build_profile.log
```

The first few lines of the resulting `build_profile.prf` will look similar to (*see* **Note 7**) the following (note that the rightmost column, containing the primary sequence, has been omitted here for clarity):

```
# Number of sequences:     51

# Length of profile  :    335

# N_PROF_ITERATIONS  :      1

# GAP_PENALTIES_1D   :   -500.0   -50.0

# MATRIX_OFFSET      : -450.0

# RR_FILE            : ${LIB}/blosum62.sim.mat

 1 TvLDH    S  0  335   1   335   0    0     0   0.  0.0

 2 1a5zA    X  1  312  75   242  63  229   164  28. 0.34E-07

 3 2a92A    X  1  316   8   191   6  186   174  26. 0.69E-04

 4 4aj2A    X  1  327  85   301  89  300   207  25. 0.15E-04

 5 1b8pA    X  1  327   7   331   6  325   316  42.  0.0
```

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 (1) the second column, containing the PDB code of the related template sequences; (2) the eleventh column, containing the percentage sequence identity between the TvLDH and template sequences; and (3) the twelfth column, containing the *E*-values for the statistical significance of the alignments. These columns are shown in bold above.

The extent of similarity between the target–template pairs is usually quantified using sequence identity or a statistical measure such as *E*-value (*see* **Note 8**). 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 nine PDB sequences, all corresponding to malate dehydrogenases (1b8pA, 1civA, 3d5tA, 4h7pA, 4h7pB, 5mdhA, 7mdhA, 1smkA, 1y7tA) that show significant similarities to TvLDH with *E*-values of zero.

3.2 Sequence–Structure Alignment

The next step is to align the target TvLDH sequence with the chosen template (*see* **Note 9**). Here, the 1y7tA template is used. This alignment is created using MODELLER's `align2d()` function (*see* **Note 10**). Although `align2d()` is based on a global dynamic programming algorithm [22], 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 not between two positions that are close in space [14]. 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 11**).

This step is carried out by running

```
python align2d.py>align2d.log
```

This script reads in the PDB structure of the template and the sequence of the target (TvLDH) and calls the `align2d()` function to perform the alignment. The resulting alignment is written out in two formats. `TvLDH-1y7tA.ali` in the PIR format is subsequently used by MODELLER for modeling; `TvLDH-1y7tA.pap` in the PAP format is easier to read, for example to see which residues are aligned with each other.

3.3 Model Building

Models of TvLDH can now be built by running

```
python model.py>model.log
```

The script uses MODELLER's `automodel` class, specifying the name of the alignment file to use and the identifiers of the target (TvLDH) and template (1y7tA) sequences. It then asks `automodel` to generate five models (*see* **Note 12**). Each is assessed with the normalized Discrete Optimized Protein Energy (DOPE) assessment method [18]. The five models are written out as PDB files with names `TvLDH.B9999[0001-0005].pdb`.

3.4 Model Evaluation

The log file produced by the model building procedure (`model.log`) contains a summary of each calculation at the bottom of the file. This summary includes, for each of the five models, the MODELLER objective function (*see* **Note 13**) [12] and the normalized DOPE score (*see* **Note 14**). These scores can be used to identify which of the five models produced is likely to be the most accurate model (*see* **Note 15**).

Since the DOPE potential is simply a sum of interactions between pairs of atoms, it can be decomposed into a score per residue, which is termed in MODELLER an "energy profile." This energy profile can be generated for the model with the best DOPE score by running the `make_energy_profile.py` script. The script outputs the profile, `TvLDH.profile`, in a simple format that is easily displayed in any graphing package. Such a profile is useful to detect local regions of high pseudo-energy that usually correspond to errors in the model (*see* **Notes 16** and **17**).

3.5 Use of Multiple Templates

One way to potentially improve the accuracy of generated models is to use multiple-template structures. When there are multiple templates, different template structures may be of higher local sequence identity to the target (or higher quality) than others in different regions, allowing MODELLER to build a model based on the most useful structural information for each region in the protein. The procedure is demonstrated here using the five templates that have the highest sequence identity to the target (1b8pA, 4h7pA, 4h7pB, 5mdhA, 1y7tA). Input files can be found in the "`multiple`" subdirectory of the zipfile. The first step is to align all of the templates with each other, which can be done by running

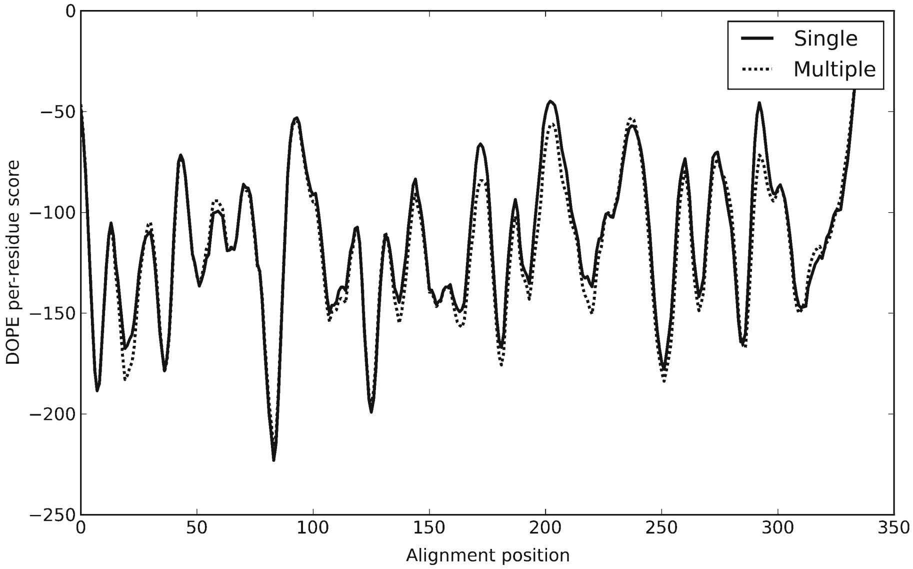

Fig. 2 The DOPE [18] energy profiles for the best-assessed model generated by modeling with a single template (*solid line*) and multiple templates (*dotted line*). Peaks (local regions of high, unfavorable score) tend to correspond to errors in the models

```
python salign.py>salign.log
```

This script uses MODELLER's `salign()` function [15] to read in all of the template structures and then generate their best structural alignment (*see* **Note 18**), written out as `templates.ali`.

Next, just as for single-template modeling, the target is aligned with the templates using the `align2d()` function. The function's `align_block` parameter is set to 5 to align the target sequence with the pre-aligned block of templates, and not to change the existing alignment between individual templates:

```
python align2d.py>align2d.log
```

Finally, model generation proceeds just as for the single-template case (the only difference is that `automodel` is now given a list of all five templates):

```
python model.py>model.log
```

Comparison of the normalized DOPE scores from the end of this logfile with those from the single-template case shows an improvement in the DOPE score of the best model from −0.92 to −1.19. Figure 2 shows the energy profiles of the best scoring models from each procedure (generated using the `plot_profiles.py` script). It can be seen that some of the predicted errors in the single-template model (peaks in the graph) have been resolved in the model calculated using multiple templates.

3.6 External Assessment

Models generated by MODELLER are stored in PDB files and so can be evaluated for accuracy with other methods if desired. One such method is the ModEval web server at http://salilab.org/evaluation/. This server takes as input the PDB file and the MODELLER PIR alignment used to generate it. It returns not only the normalized DOPE score and the energy profile but also the GA341 assessment score [23, 24] and an estimate of the Cα RMSD and native overlap between the model and its hypothetical native structure, using the TSVMod method [25]; native overlap is defined as the fraction of Cα atoms in the model that are within 3.5 Å of the same Cα atom in the native structure after least squares superposition.

3.7 Structures of Complexes

The example shown here generates a model of a single protein. However, MODELLER can also generate models of complexes of multiple proteins if templates for the entire complex are available; examples can be found in the MODELLER manual. In the case where only templates for the individual subunits in the complex are available, comparative models can be docked in a pairwise fashion by molecular docking [26, 27] or assembled based on various experimental data to generate approximate models of the complex using a wide variety of integrative modeling methods [28–31]. For example, if a cryo-electron microscopy density map of the complex is available, a model of the whole complex can be constructed by simultaneously fitting comparative models of the subunits into the density map using the MultiFit method [32] or its associated web server at http://salilab.org/multifit/ [33]. Alternatively, if a small-angle X-ray (SAXS) profile of a dimer is available, models of the dimer can be generated by docking the two subunits, constrained by the SAXS data, using the FoXSDock web server at http://salilab.org/foxsdock/ [34]. Both of these methods are part of the open-source *Integrative Modeling Platform* (IMP) package [29].

4 Notes

1. The MODELLER website also contains a full manual, a mailing list, and more example MODELLER scripts. A license key is required to use MODELLER, but this can also be obtained from the website.
2. The sequence identity is a useful 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) for Cα atoms of less than 3.5 Å (Fig. 3). Sequence–structure relationships in the "twilight zone" [35] (corresponding to relationships with statistically significant sequence similarity with identities

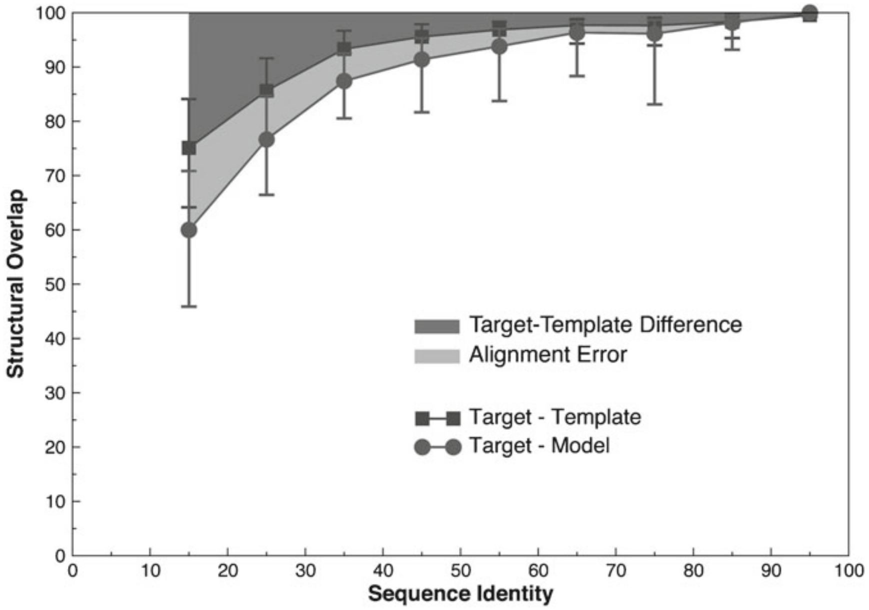

Fig. 3 Average model accuracy as a function of sequence identity [54]. As the sequence identity between the target sequence and the template structure decreases, the average structural similarity between the template and the target also decreases (*dark grey area*, *squares*) [55]. Structural overlap is defined as the fraction of equivalent Cα atoms. For the comparison of the model with the actual structure (*circles*), two Cα 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 Cα 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

generally in the 10–30 % range), or the “midnight zone” [35] (corresponding to statistically insignificant sequence similarity), typically result in less accurate models.

3. The database contains sequences of the structures from PDB. To increase the search speed, redundancy is removed from the database; the PDB sequences are clustered with other sequences that are at least 95 % identical, and only the representative of each cluster is stored in the database. This database is termed “pdb_95.” A copy of this database is included in the downloaded zipfile as `pdb_95.pir`. Newer versions of this database, updated as new structures are deposited in PDB, can be downloaded from the MODELLER website at http://salilab.org/modeller/supplemental.html.

4. MODELLER is a command line tool, and so all commands must be run by typing at the command line. All of the necessary

input files for this demonstration are in the downloaded zipfile; simply download and extract the zipfile and change into the newly created directory (using the "`cd`" command at the command line). After this, MODELLER scripts can be run as shown in the text. All MODELLER scripts are Python scripts and so should be run with the "`python`" command. (On some systems the full path to the Python interpreter is necessary, such as `/usr/bin/python` on a Linux or a Mac machine or `C:\python26\python.exe` on a Windows system.) MODELLER scripts can also be run from other Python frontends, such as IDLE, if desired. On a Windows system, it is generally *not* a good idea to simply "double click" on a MODELLER Python script, since any output from the script will disappear as soon as it finishes. Finally, if Python is not installed, MODELLER includes a basic Python 2.3 interpreter as "`mod<version>`." For example, to run this first script using MODELLER version 9.11's own interpreter, run "`mod9.11 make_pdb_95.py`." Note that `mod9.11` automatically creates a "`make_pdb_95.log`" logfile.

5. The binary database is much faster to use than the original text format database, `pdb_95.pir`. Note, however, that it is not necessarily smaller. This script does not need to be run again unless `pdb_95.pir` is updated.

6. `TvLDH.ali` simply contains the primary sequence of the target, in MODELLER's variant of the PIR format (which is documented in more detail in the MODELLER manual). This file is included in the zipfile.

7. Although MODELLER's algorithms are deterministic, exactly the same job run on different machines (e.g., a Linux box versus a Windows or a Mac machine) may give different results. This difference may arise because different machines handle rounding of floating point numbers and ordering of floating point operations differently, and the minor differences introduced can be compounded and end up giving very different outputs. This variation is normal and to be expected, and so the results shown in this text may differ from those obtained by running MODELLER elsewhere.

8. The sequence identity is not a statistically reliable measure of alignment significance and corresponding model accuracy for values lower than 30 % [35, 36]. 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 [37], that compare the score obtained for an alignment with an established background distribution of such scores. 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 [36]. 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 corrected for the length of the aligned segment before the significance is computed [37, 38].

9. 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, pH, and quaternary contacts.

10. 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.

11. Most alignment methods use either the local or the 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) [14], the substitution matrix used to score the aligned residues (20 × 20 matrices derived from alignments with a given sequence identity, those derived from structural alignments, and those incorporating the structural environment of the residues) [39], or combinations of both [40–43]. There does not 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 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 [44, 45].

12. To generate each model, MODELLER takes a starting structure, which is simply the target sequence threaded onto the template backbone, adds some randomization to the coordinates, and then optimizes it by searching for the minimum of its scoring function. Since finding the global minimum of the scoring function is not guaranteed, it is usually recommended to repeat the procedure multiple times to generate an ensemble of models; the randomization is necessary otherwise, the same model would be generated each time. Computing multiple models is particularly important when the sequence–structure alignment contains different templates with many insertions and/or deletions. Calculating multiple models allows for better sampling of the different template segments and the conformations of the unaligned regions. The best scoring model among these multiple models is generally more accurate than the first model produced.

13. 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 [12].

14. The DOPE score [18] is an atomic distance-dependent statistical potential based on a physical reference state that accounts for the finite size and spherical shape of proteins. The reference state assumes that 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 predicting more accurate models. Since DOPE is a pseudo-energy dependent on the composition and size of the system, DOPE scores are only directly comparable for models with the same set of atoms (so can, for example, be used to rank multiple models of the same protein, but cannot be used without additional approximations to compare models of a protein and its mutant). The normalized DOPE (or z-DOPE) score, however, is a *z* score that relates the DOPE score of the model to the average observed DOPE

score for "reference" protein structures of similar size [25]. Negative normalized DOPE scores of −1 or below are likely to correspond to models with the correct fold.

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 atomic distance statistical potentials have been shown to have the greatest ability to differentiate models in the ~3 Å Cα RMSD range. Tests show that such scores are often able to identify a model within 0.5 Å Cα RMSD of the most accurate model produced [46]. 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 (i.e., insertions or loops) are among the most difficult regions to model [11, 47–49]. 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 incorporate structural information can often correct such errors [14]. Once a reliable alignment is obtained, various modeling protocols can predict the loop conformation, for insertions of less than approximately ten residues long [11, 47, 50, 51].

17. As a consequence of sequence divergence, the main-chain 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 (e.g., packing of subunits in a crystal and ligands). The simultaneous use of several templates can minimize this kind of error [52, 53].

18. It is particularly important to generate the best alignment of the structures to minimize conflicting information (e.g., one template suggesting that two Cα atoms in the target are close and another suggesting they are widely separated). SALIGN [15] uses both sequence- and structure-dependent features to align multiple structures. It employs an iterative procedure to determine the input parameters that maximize the structural overlap of the generated alignment.

Acknowledgments

We are grateful to all members of our research group. We also acknowledge support from National Institutes of Health (U54 GM094625) as well as computing hardware support from Ron Conway, Mike Homer, Hewlett-Packard, NetApp, IBM, and Intel.

References

1. Baker D, Sali A (2001) Protein structure prediction and structural genomics. Science 294(5540):93–96
2. Schwede T, Sali A, Honig B et al (2009) Outcome of a workshop on applications of protein models in biomedical research. Structure 17(2):151–159
3. Zhang Y (2008) Progress and challenges in protein structure prediction. Curr Opin Struct Biol 18(3):342–348
4. Marti-Renom MA, Stuart AC, Fiser A et al (2000) Comparative protein structure modeling of genes and genomes. Annu Rev Biophys Biomol Struct 29:291–325
5. Eswar N, Sali A (2009) Protein structure modeling. In: Sussman JL, Spadon P (eds) From molecules to medicine, structure of biological macromolecules and its relevance in combating new diseases and bioterrorism, NATO science for peace and security series: a—chemistry and biology. Springer, Dordrecht, The Netherlands, pp 139–151
6. Ginalski K (2006) Comparative modeling for protein structure prediction. Curr Opin Struct Biol 16(2):172–177
7. Das R, Baker D (2008) Macromolecular modeling with rosetta. Annu Rev Biochem 77: 363–382
8. Zhang Y, Skolnick J (2004) Automated structure prediction of weakly homologous proteins on a genomic scale. Proc Natl Acad Sci USA 101(20):7594–7599
9. Simons KT, Bonneau R, Ruczinski I, et al (1999) Ab initio protein structure prediction of CASP III targets using ROSETTA. Proteins (Suppl 3):171–176
10. Pieper U, Webb BM, Barkan DT et al (2011) ModBase, a database of annotated comparative protein structure models, and associated resources. Nucleic Acids Res 39:465–474
11. Fiser A, Do RKG, Sali A (2000) Modeling of loops in protein structures. Protein Sci 9(9): 1753–1773
12. Sali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 234(3):779–815
13. Marti-Renom MA, Madhusudhan MS, Sali A (2004) Alignment of protein sequences by their profiles. Protein Sci 13(4):1071–1087
14. Madhusudhan MS, Marti-Renom MA, Sanchez R et al (2006) Variable gap penalty for protein sequence-structure alignment. Protein Eng Des Sel 19(3):129–133
15. Madhusudhan MS, Webb BM, Marti-Renom MA et al (2009) Alignment of multiple protein structures based on sequence and structure features. Protein Eng Des Sel 22:569–574
16. Brooks BR, Brooks CL 3rd, Mackerell AD Jr et al (2009) CHARMM: the biomolecular simulation program. J Comput Chem 30(10): 1545–1614
17. Sali A, Overington JP (1994) Derivation of rules for comparative protein modeling from a database of protein structure alignments. Protein Sci 3(9):1582–1596
18. Shen MY, Sali A (2006) Statistical potential for assessment and prediction of protein structures. Protein Sci 15(11):2507–2524
19. Wu G, Fiser A, ter Kuile B et al (1999) Convergent evolution of Trichomonas vaginalis lactate dehydrogenase from malate dehydrogenase. Proc Natl Acad Sci USA 96(11): 6285–6290
20. Berman HM, Westbrook J, Feng Z et al (2000) The protein data bank. Nucleic Acids Res 28(1):235–242
21. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147(1):195–197
22. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453
23. John B, Sali A (2003) Comparative protein structure modeling by iterative alignment, model building and model assessment. Nucleic Acids Res 31(14):3982–3992
24. Melo F, Sanchez R, Sali A (2002) Statistical potentials for fold assessment. Protein Sci 11(2):430–448
25. Eramian D, Eswar N, Shen M et al (2008) How well can the accuracy of comparative

protein structure models be predicted? Protein Sci 17(11):1881–1893

26. Vajda S, Kozakov D (2009) Convergence and combination of methods in protein–protein docking. Curr Opin Struct Biol 19(2):164–170
27. Lensink MF, Wodak SJ (2010) Docking and scoring protein interactions: CAPRI 2009. Proteins 78(15):3073–3084
28. Alber F, Forster F, Korkin D et al (2008) Integrating diverse data for structure determination of macromolecular assemblies. Annu Rev Biochem 77:443–477
29. Russel D, Lasker K, Webb B et al (2012) Putting the pieces together: integrative structure determination of macromolecular assemblies. PLoS Biol 10(1):e1001244
30. Robinson C, Sali A, Baumeister W (2007) The molecular sociology of the cell. Nature 450(7172):973–982
31. Ward A, Sali A, Wilson I (2013) Structural biology unleashed. Science 339:913–915
32. Lasker K, Sali A, Wolfson HJ (2010) Determining macromolecular assembly structures by molecular docking and fitting into an electron density map. Proteins Struct Funct Bioinform 78:3205–3211
33. Tjioe E, Lasker K, Webb B et al (2011) MultiFit: a web server for fitting multiple protein structures into their electron microscopy density map. Nucleic Acids Res 39:167–170
34. Schneidman-Duhovny D, Hammel M, Sali A (2011) Macromolecular docking restrained by a small angle X-ray scattering profile. J Struct Biol 3:461–471
35. Rost B (1999) Twilight zone of protein sequence alignments. Protein Eng 12(2): 85–94
36. May AC (2004) Percent sequence identity; the need to be explicit. Structure 12(5):737–738
37. Altschul SF, Madden TL, Schaffer AA et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25(17):3389–3402
38. Pearson WR (1998) Empirical statistical estimates for sequence similarity searches. J Mol Biol 276(1):71–84
39. Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA 89(22):10915–10919
40. Zhou H, Zhou Y (2005) Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. Proteins 58(2): 321–328
41. McGuffin LJ, Jones DT (2003) Improvement of the GenTHREADER method for genomic fold recognition. Bioinformatics 19(7): 874–881
42. Karchin R, Cline M, Mandel-Gutfreund Y et al (2003) Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins 51(4):504–514
43. Shi J, Blundell TL, Mizuguchi K (2001) FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. J Mol Biol 310(1):243–257
44. Dunbrack RL Jr (2006) Sequence comparison and protein structure prediction. Curr Opin Struct Biol 16(3):374–384
45. Xiang Z (2006) Advances in homology protein structure modeling. Curr Protein Pept Sci 7(3):217–227
46. Eramian D, Shen M, Devos D et al (2006) A composite score for predicting errors in protein structure models. Protein Sci 15(7):1653–1666
47. Jacobson MP, Pincus DL, Rapp CS et al (2004) A hierarchical approach to all-atom protein loop prediction. Proteins 55(2):351–367
48. Zhao S, Zhu K, Li J et al (2011) Progress in super long loop prediction. Proteins 79(10): 2920–2935
49. Fernandez-Fuentes N, Oliva B, Fiser A (2006) A supersecondary structure library and search algorithm for modeling loops in protein structures. Nucleic Acids Res 34(7):2085–2097
50. van Vlijmen HW, Karplus M (1997) PDB-based protein loop prediction: parameters for selection and methods for optimization. J Mol Biol 267(4):975–1001
51. Coutsias EA, Seok C, Jacobson MP et al (2004) A kinematic view of loop closure. J Comput Chem 25(4):510–528
52. Sanchez R, Sali A (1997) Evaluation of comparative protein structure modeling by MODELLER-3. Proteins (Suppl 1):50–58
53. Srinivasan N, Blundell TL (1993) An evaluation of the performance of an automated procedure for comparative modelling of protein tertiary structure. Protein Eng 6(5):501–512
54. Sanchez R, Sali A (1998) Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. Proc Natl Acad Sci USA 95(23):13597–13602
55. Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. EMBO J 5(4):823–826