Comparative protein structure modeling with MODELLER: A practical approach

András Fiser and Andrej Šali

Laboratories of Molecular Biophysics

Pels Family Center for Biochemistry and Structural Biology

The Rockefeller University

1230 York Avenue, New York, NY 10021, USA

Correspondence to Andrej Šali

The Rockefeller University

1230 York Avenue, New York, NY 10021, USA

tel: +1 (212) 327 7550; fax: +1 (212) 327 7540

e-mail: sali@rockefeller.edu

Running title: Comparative protein structure modeling

August 7, 2001

1 Introduction

Functional characterization of a protein sequence is one of the most frequent problems in biology. This task is usually facilitated by accurate three-dimensional (3D) structure of the studied protein. In the absence of an experimentally determined structure, comparative or homology modeling can sometimes provide a useful 3D model for a protein (target) that is related to at least one known protein structure (template) [1–7].

Despite progress in *ab initio* protein structure prediction [8], comparative modeling remains the only method that can reliably predict the 3D structure of a protein with an accuracy comparable to a low-resolution experimentally determined structure [6]. Even models with errors may be useful, because some aspects of function can be predicted from only coarse structural features of a model. Typical uses of comparative models are listed in Table 1 [4, 6].

3D structure of proteins from the same family is more conserved than their primary sequences [9]. Therefore, if similarity between two proteins is detectable at the sequence level, structural similarity can usually be assumed. Moreover, proteins that share low or even non-detectable sequence similarity many times also have similar structures. Currently, the probability to find related proteins of known structure for a sequence picked randomly from a genome ranges approximately from 20% to 65%, depending on the genome [10, 11]. Approximately one half of all known sequences have at least one domain that is detectably related to at least one protein of known structure [10]. Since the number of known protein sequences is approximately 600,000 [12, 13], comparative modeling can be applied to domains in approximately 300,000 proteins. This number is an order of magnitude larger than the number of experimentally determined protein structures deposited in the Protein Data Bank (PDB) ($\sim 15,000$) [14]. Furthermore, the usefulness of comparative modeling is steadily increasing because the number of different structural folds that proteins adopt is limited [15–18] and because the number of experimentally determined new structures is increasing exponentially [19]. This trend is accentuated by the recently initiated structural genomics project

that aims to determine at least one structure for most protein families [20, 21]. It is conceivable that this aim will be substantially achieved in less than 10 years, making comparative modeling applicable to most protein sequences.

Comparative modeling usually consists of the following five steps: search for related protein structures, selection of one or more templates, target–template alignment, model building, and model evaluation (Figure 1). If the model is not satisfactory, some or all of the steps can be repeated.

There are several computer programs and web servers that automate the comparative modeling process. The first web server for automated comparative modeling was the Swiss-Model server (`http://www.expasy.ch/swissmod/`), followed by CPHModels (`http://www.cbs.dtu.dk/services/CPHmodels/`), SDSC1 (`http://cl.sdsc.edu/hm`), FAMS (`http://physchem.pharm.kitasato-u.ac.jp/FAMS/fams.html`) and ModWeb (`http://guitar.rockefeller.edu/modweb/`). These servers accept a sequence from a user and return an all atom comparative model when possible. In addition to modeling a given sequence, ModWeb is also capable of returning comparative models for all sequences in the TrEMBL database that are detectably related to an input, user provided structure. While the web servers are convenient and useful, the best results in the difficult or unusual modeling cases, such as problematic alignments, modeling of loops, existence of multiple conformational states, and modeling of ligand binding, are still obtained by non-automated, expert use of the various modeling tools. A number of resources useful in comparative modeling are listed in Table 2.

Next, we describe generic considerations in all five steps of comparative modeling (Section 2). We then illustrate these considerations in practice by discussing three applications of our program Modeller [22–24] to specific modeling problems (Section 3). This chapter does not review the comparative modeling field in general [6].

2 Comparative modeling steps

2.1 Searching for structures related to the target sequence

Comparative modeling usually starts by searching the Protein Data Bank (PDB) of known protein structures using the target sequence as the query. This search is generally done by comparing the target sequence with the sequence of each of the structures in the database. A variety of sequence–sequence comparison methods can be used [25–29]. Frequently, availability of many sequences related to the target or potential templates allows more sensitive searching with sequence profile methods and Hidden Markov Models [30–34]. Another kind of a search is based on evaluating the compatibility between the target sequence and each of the structures in the database, achieved by the "threading" group of methods [35–40]. Threading uses sequence–structure fitness functions, such as residue-level statistical potential functions, to evaluate a sequence–structure match. Threading methods generally do not rely on sequence similarity. Threading sometimes detects structural similarity between proteins without detectable sequence similarity [41].

A good starting point for template searches are the many database search servers on the Internet (Table 2). The most useful ones are those that search directly against the PDB, such as PDB-Blast (`http://bioinformatics.burnham-inst.orgpdb_blast`). When the target sequence is only remotely related to known structures, it is frequently useful to try several different methods for finding related structures.

2.2 Selecting templates

Once a list of potential templates is obtained using searching methods, it is necessary to select one or more templates that are appropriate for the particular modeling problem. Several factors need to be taken into account when selecting a template.

The quality of a template increases with its overall sequence similarity to the target and de-

creases with the number and length of gaps in the alignment. The simplest template selection rule is to select the structure with the higher sequence similarity to the modeled sequence.

The family of proteins that includes the target and the templates can frequently be organized into sub-families. The construction of a multiple alignment and a phylogenetic tree [42] can help in selecting the template from the subfamily that is closest to the target sequence.

The similarity between the "environment" of the template and the environment in which the target needs to be modeled should also be considered. The term "environment" is used here in a broad sense, including everything that is not the protein itself (*e.g.*, solvent, *p*H, ligands, quaternary interactions). If possible, a template bound to the same or similar ligands as the modeled sequence should generally be used.

The quality of the experimentally determined structure is another important factor in template selection. Resolution and R-factor of a crystallographic structure and the number of restraints per residue for an NMR structure are indicative of the accuracy of the structure. This information can generally be obtained from the template PDB files or from the articles describing structure determination. For instance, if two templates have comparable sequence similarity to the target, the one determined at the highest resolution should generally be used.

The criteria for selecting templates also depend on the purpose of a comparative model. For example, if a protein–ligand model is to be constructed, the choice of the template that contains a similar ligand is probably more important than the resolution of the template. On the other hand, if the model is to be used to analyze the geometry of the active site of an enzyme, it may be preferable to use a high-resolution template structure.

It is not necessary to select only one template. In fact, the use of several templates generally increases the model accuracy. One strength of MODELLER is that it can combine information from multiple template structures, in two ways. First, multiple template structures may be aligned with different domains of the target, with little overlap between them, in which case the modeling pro-

cedure can construct a homology-based model of the whole target sequence. Second, the template structures may be aligned with the same part of the target, in which case the modeling procedure is likely to automatically build the model on the locally best template [43, 44]. In general, it is frequently beneficial to include in the modeling process all the templates that differ substantially from each other, if they share approximately the same overall similarity to the target sequence.

An elaborate way to select suitable templates is to generate and evaluate models for each candidate template structure and/or their combinations. The optimized all-atom models are evaluated by an energy or scoring function, such as the Z-score of ProsaII [45]. The ProsaII Z-score of a model is a measure of compatibility between its sequence and structure. Ideally, the Z-score of the model should be comparable to the Z-score of the template. ProsaII Z-score is frequently sufficiently accurate to allow picking one of the most accurate of the generated models [46]. This trial-and-error approach can be viewed as limited threading (*i.e.*, the target sequence is threaded through similar template structures). For additional comments on model assessment see Section 2.5.

2.3 Aligning the target sequence with one or more structures

To build a model, all comparative modeling programs depend on a list of assumed structural equivalences between the target and template residues. This list is defined by the alignment of the target and template sequences. Although many template search methods will produce such an alignment, it is usually not the optimal target–template alignment in the more difficult alignment cases (*e.g.*, at less than 30% sequence identity). Search methods tend to be tuned for detection of remote relationships, not for optimal alignment. Therefore, once the templates are selected, an alignment method should be used to align them with the target sequence. The alignment is relatively simple to obtain when the target–template sequence identity is above 40%. In most such cases, an accurate alignment can be obtained automatically using standard sequence–sequence alignment methods. If the target–template sequence identity is lower than 40%, the alignment

generally has gaps and needs manual intervention to minimize the number of misaligned residues. In these low sequence identity cases, the alignment accuracy is the most important factor affecting the quality of the resulting model. Alignments can be improved by including structural information from the template. For example, gaps should be avoided in secondary structure elements, in buried regions, or between two residues that are far in space. Some alignment methods take such criteria into account [11, 47–50]. It is important to inspect and edit the alignment in view of the template structure, especially if the target–template sequence identity is low. A misalignment by only one residue position will result in an error of approximately 4Å in the model because the current modeling methods generally cannot recover from errors in the alignment.

When multiple templates are selected, a good strategy is to superpose them with each other first, to obtain a multiple structure-based alignment. In the next step, the target sequence is aligned with this multiple structure-based alignment. Another improvement is to calculate the target and template sequence profiles, by aligning them with all sequences from a non-redundant sequence database that are sufficiently similar to the target and template sequences, respectively, so that they can be aligned without significant errors (*e.g.*, better than 40% sequence identity). The final target–template alignment is then obtained by aligning the two profiles, not the template and target sequences alone. The use of multiple structures and multiple sequences benefits from the evolutionary and structural information about the templates as well as evolutionary information about the target sequence, and often produces a better alignment for modeling than the pairwise sequence alignment methods [51, 52].

2.4 Model Building

Once an initial target–template alignment is built, a variety of methods can be used to construct a 3D model for the target protein [1–6]. The original and still widely used method is modeling by rigid-body assembly [1, 53, 54]. This method constructs the model from a few core regions and

from loops and sidechains, which are obtained from dissecting related structures. Another family of methods, modeling by segment matching, relies on the approximate positions of conserved atoms from the templates to calculate the coordinates of other atoms [55–58]. The third group of methods, modeling by satisfaction of spatial restraints, uses either distance geometry or optimization techniques to satisfy spatial restraints obtained from the alignment of the target sequence with the template structures [22, 59–62]. Specifically, MODELLER, which belongs to this group of methods, extracts spatial restraints from two sources. First, homology-derived restraints on the distances and dihedral angles in the target sequence are extracted from its alignment with the template structures. Second, stereochemical restraints such as bond length and bond angle preferences are obtained from the molecular mechanics force field of CHARMM-22 [63] and statistical preferences of dihedral angles and non-bonded atomic distances are obtained from a representative set of all known protein structures. The model is then calculated by an optimization method relying on conjugate gradients and molecular dynamics, which minimizes violations of the spatial restraints (Figure 2). The procedure is conceptually similar to that used in determination of protein structures from NMR-derived restraints. The fourth group of comparative model building methods starts with an alignment and then searches the conformational space guided by a statistical potential function and somewhat relaxed homology restraints derived from the input alignment, in an attempt to overcome at least some alignment mistakes [64].

Accuracies of the various model building methods are relatively similar when used optimally. Other factors such as template selection and alignment accuracy usually have a larger impact on the model accuracy, especially for models based on less than 40% sequence identity to the templates. However, it is important that a modeling method allows a degree of flexibility and automation to obtain better models more easily and rapidly. For example, a method should allow for an easy recalculation of a model when a change is made in the alignment; it should be straightforward to calculate models based on several templates; and the method should provide tools for incorporation of prior knowledge about the target (*e.g.*, cross-linking restraints, predicted secondary structure)

and allow *ab initio* modeling of insertions (*e.g.*, loops), which can be crucial for annotation of function. Loop modeling is an especially important aspect of comparative modeling in the range from 30 to 50% sequence identity. In this range of overall similarity, loops among the homologs vary while the core regions are still relatively conserved and aligned accurately. Next, we single out loop modeling and review it in more detail.

There are two approaches to loop modeling. First, the *ab initio* loop prediction is based on a conformational search or enumeration of conformations in a given environment, guided by a scoring or energy function. There are many such methods, exploiting different protein representations, energy function terms, and optimization or enumeration algorithms [24]. The second, database approach to loop prediction consists of finding a segment of mainchain that fits the two stem regions of a loop. The search for such a segment is performed through a database of many known protein structures, not only homologs of the modeled protein. Usually, many different alternative segments that fit the stem residues are obtained, and possibly sorted according to geometric criteria or sequence similarity between the template and target loop sequences. The selected segments are then superposed and annealed on the stem regions. These initial crude models are often refined by optimization of some energy function.

The loop modeling module in MODELLER implements the optimization-based approach [24]. The main reasons are the generality and conceptual simplicity of energy minimization, as well as the limitations on the database approach imposed by a relatively small number of known protein structures [65]. Loop prediction by optimization is applicable to simultaneous modeling of several loops and loops interacting with ligands, which is not straightforward for the database search approaches. Loop optimization in MODELLER relies on conjugate gradients and molecular dynamics with simulated annealing. The pseudo energy function is a sum of many terms, including some terms from the CHARMM-22 molecular mechanics force field [63] and spatial restraints based on distributions of distances [66] and dihedral angles [67] in known protein structures. The method was tested on a large number of loops of known structure, both in the native and near-native

environments. Loops of 8 residues predicted in the native environment have a 90% chance to be modeled with useful accuracy (*i.e.*, RMSD for superposition of the loop mainchain atoms is less than 2Å). Even 12-residue loops are modeled with useful accuracy in 30% of the cases. When the RMSD distortion of the environment atoms is 2.5Å, the average loop prediction error increases by 180, 25 and 3% for 4, 8 and 12-residue loops, respectively. It is not anymore too optimistic to expect useful models for loops as long as 12 residues, if the environment of the loop is at least approximately correct. It is possible to estimate whether or not a given loop prediction is correct, based on the structural variability of the independently derived lowest energy loop conformations.

2.5 Evaluating a model

After a model is built, it is important to check it for possible errors. The quality of a model can be approximately predicted from the sequence similarity between the target and the template (Figure 3). Sequence identity above 30% is a relatively good predictor of the expected accuracy of a model. However, other factors, including the environment, can strongly influence the accuracy of a model. For instance, some calcium-binding proteins undergo large conformational changes when bound to calcium. If a calcium-free template is used to model the calcium-bound state of a target, it is likely that the model will be incorrect irrespective of the target–template similarity. This estimate also applies to determination of protein structure by experiment; a structure must be determined in the functionally meaningful environment. If the target–template sequence identity falls below 30%, the sequence identity becomes significantly less reliable as a measure of expected accuracy of a single model. The reason is that below 30% sequence identity, models are often obtained that deviate significantly, in both directions, from the average accuracy. It is in such cases that model evaluation methods are most informative.

Two types of evaluation can be carried out. "Internal" evaluation of self-consistency checks whether or not a model satisfies the restraints used to calculate it. "External" evaluation relies on

information that was not used in the calculation of the model [45, 68].

Assessment of model's stereochemistry (*e.g.*, bonds, bond angles, dihedral angles, and non-bonded atom–atom distances) with programs such as PROCHECK [69] and WHATCHECK [70] is an example of internal evaluation. Although errors in stereochemistry are rare and less informative than errors detected by methods for external evaluation, a cluster of stereochemical errors may indicate that the corresponding region also contains other larger errors (*e.g.*, alignment errors).

When the model is based on less than ∼30% sequence identity to the template, the first purpose of the external evaluation is to test whether or not a correct template was used. This test is especially important when the alignment is only marginally significant or several alternative templates with different folds are to be evaluated. A complication is that at low similarities the alignment generally contains many errors, making it difficult to distinguish between an incorrect template on one hand and an incorrect alignment with a correct template on the other hand. It is generally possible to recognize a correct template only if the alignment is at least approximately correct. This complication can sometimes be overcome by testing models from several alternative alignments for each template. One way to predict whether or not a template is correct is to compare the PROSAII Z-score [45] for the model and the template structure(s). Since the Z-score of a model is a measure of compatibility between its sequence and structure, the model Z-score should be comparable to that of the template. However, this evaluation does not always work. For example, a well modeled part of a domain is likely to have a bad Z-score because some interactions that stabilize the fold are not present in the model. Correct models for some membrane proteins and small disulfide-rich proteins also tend to be evaluated incorrectly, apparently because these structures have distributions of residue accessibility and residue–residue distances that are different from those for the larger globular domains, which were the source of the PROSAII statistical potential functions.

The second, more detailed kind of external evaluation is the prediction of unreliable regions in the model. One way to approach this problem is to calculate a "pseudo energy" profile of a model,

such as that produced by ProsaII. The profile reports the energy for each position in the model. Peaks in the profile frequently correspond to errors in the model. There are several pitfalls in the use of energy profiles for local error detection. For example, a region can be identified as unreliable only because it interacts with an incorrectly modeled region; there are also more fundamental problems [24].

Finally, a model should be consistent with experimental observations, such as site-directed mutagenesis, cross-linking data, and ligand binding.

Are comparative models "better" than their templates? In general, models are as close to the target structure as the templates, or slightly closer if the alignment is correct [44]. This is not a trivial achievement because of the many residue substitutions, deletions and insertions that occur when the sequence of one protein is transformed into the sequence of another. Even in a favorable modeling case with a template that is 50% identical to the target, half of the sidechains change and have to be packed in the protein core such that they avoid atom clashes and violations of stereochemical restraints. When more than one template is used for modeling, it is sometimes possible to obtain a model that is significantly closer to the target structure than any of the templates [43, 44]. This improvement occurs because the model tends to inherit the best regions from each template. Alignment errors are the main factor that may make models worse than the templates. However, to represent the target, it is always better to use a comparative model rather than the template. The reason is that the errors in the alignment affect similarly the use of the template as a representation of the target as well as a comparative model based on that template [44].

2.6 Iterating alignment, modeling and model evaluation

It is frequently difficult to select best templates or calculate a good alignment. One way of improving a comparative model in such cases is to proceed with an iteration consisting of template selection,

alignment, and model building, guided by model assessment. This iteration can be repeated until no improvement in the model is detected [44, 71].

3 Modeling examples using MODELLER

This section contains three examples of a typical comparative modeling application. All the examples use program MODELLER-6 and other freely available software. The first example demonstrates each of the five steps of comparative modeling at their most basic level. The second example illustrates the use of multiple templates and modeling of a protein with a ligand and a co-factor, as well as applying user-defined restraints for docking a substrate molecule into the active site pocket. In the third example, we describe a loop modeling exercise. All the input and output files for MODELLER-6 can be downloaded from `http://guitar.rockefeller.edu/modeller/methenz/`. For more information, the MODELLER manual [72] and literature [22–24, 43, 44] can be consulted. A list of our papers using MODELLER to address practical problems in collaboration with experimentalists can be obtained at URL `http://guitar.rockefeller.edu/modeller/methenz/`.

Although the main purpose of MODELLER is model building, it can be used in all stages of comparative modeling, including template search, template selection, target–template alignment, model building, and model assessment. Once a target–template alignment is obtained, the calculation of a 3D model of the target by MODELLER is completely automated.

3.1 Example 1: Modeling lactate dehydrogenase from *Trichomonas vaginalis* based on a single template

A novel gene for lactate dehydrogenase was identified from the genomic sequence of *Trichomonas vaginalis* (TvLDH). The corresponding protein had a higher similarity to the malate dehydrogenase of the same species (TvMDH) than to any other LDH. We hypothesized that TvLDH arose from

TvMDH by convergent evolution relatively recently [73]. Comparative models were constructed for TvLDH and TvMDH to study the sequences in the structural context and to suggest site-directed mutagenesis experiments for elucidating specificity changes in this apparent case of convergent evolution of enzymatic specificity. The native and mutated enzymes were expressed and their activities were compared [73]. The individual modeling steps of this study are described next.

3.1.1 Searching for structures related to TvLDH

First, it is necessary to put the target TvLDH sequence into the PIR format [74] readable by MODELLER(file `'TvLDH.ali'`).

```
>P1;TvLDH
sequence:TvLDH:::::::0.00: 0.00
MSEAAHVLITGAAGQIGYILSHWIASGELYG-DRQVYLHLLDIPPAMNRLTALTMELEDCAFPHLAGFVATTDPK
AAFKDIDCAFLVASMPLKPGQVRADLISSNSVIFKNTGEYLSKWAKPSVKVLVIGNPDNTNCEIAMLHAKNLKPE
NFSSLSMLDQNRAYYEVASKLGVDVKDVHDIIVWGNHGESMVADLTQATFTKEGKTQKVVDVLDHDYVFDTFFKK
IGHRAWDILEHRGFTSAASPTKAAIQHMKAWLFGTAPGEVLSMGIPVPEGNPYGIKPGVVFSFPCNVDKEGKIHV
VEGFKVNDWLREKLDFTEKDLFHEKEIALNHLAQGG*
```

The first line contains the sequence code, in the format `'>P1;code'`. The second line with ten fields separated by colons generally contains information about the structure file, if applicable. Only two of these fields are used for sequences, `'sequence'` (indicating that the file contains a sequence without known structure) and `'TvLDH'` (the model file name). The rest of the file contains the sequence of TvLDH, with `'*'` marking its end. A search for potentially related sequences of known structure can be performed by the **SEQUENCE_SEARCH** command of MODELLER. The following script uses the query sequence `'TvLDH'` assigned to the variable ALIGN_CODES from the file `'TvLDH.ali'` assigned to the variable FILE (file `'seqsearch.top'`).

```
SET SEARCH_RANDOMIZATIONS = 100
SET FILE = 'TvLDH.ali'
SEQUENCE_SEARCH ALIGN_CODES = 'TvLDH', DATA_FILE  = ON
```

The **SEQUENCE_SEARCH** command has many options [72], but in this example only

SEARCH_RANDOMIZATIONS and DATA_FILE are set to non-default values. SEARCH_RANDOMIZATIONS specifies the number of times the query sequence is randomized during the calculation of the significance score for each sequence–sequence comparison. The higher the number of randomizations, the more accurate the significance score. DATA_FILE = ON triggers creation of an additional summary output file (`'seqsearch.dat'`).

3.1.2 Selecting a template

The output of the `'search.top'` script is written to the `'search.log'` file. MODELLER always produces a log file. Errors and warnings in log files can be found by searching for the `'_E>'` and `'_W>'` strings, respectively. At the end of the log file, MODELLER lists the hits sorted by alignment significance. Because the log file is sometimes very long, a separate data file is created that contains the summary of the search. The example shows only the top 10 hits (file `'search.dat'`).

```
  # CODE_1      CODE_2  LEN1 LEN2 NID  %ID1  %ID2    SCORE  SIGNI
-------------------------------------------------------------------
   1 TvLDH      1bdmA    335  318 153  45.7  48.1  212557.   28.9
   2 TvLDH      1lldA    335  313 103  30.7  32.9  183190.   10.1
   3 TvLDH      1ceqA    335  304  95  28.4  31.3  179636.    9.2
   4 TvLDH      2hlpA    335  303  86  25.7  28.4  177791.    8.9
   5 TvLDH      1ldnA    335  316  91  27.2  28.8  180669.    7.4
   6 TvLDH      1hyhA    335  297  88  26.3  29.6  175969.    6.9
   7 TvLDH      2cmd     335  312 108  32.2  34.6  182079.    6.6
   8 TvLDH      1db3A    335  335  91  27.2  27.2  181928.    4.9
   9 TvLDH      9ldtA    335  331  95  28.4  28.7  181720.    4.7
  10 TvLDH      1cdb     335  105  69  20.6  65.7   80141.    3.8
```

The most important columns in the **SEQUENCE_SEARCH** output are the `'CODE_2'`, `'%ID'` and `'SIGNI'` columns. The `'CODE_2'` column reports the code of the PDB sequence that was compared with the target sequence. The PDB code in each line is the representative of a group of PDB sequences that share 40% or more sequence identity to each other and have less than 30 residues or 30% sequence length difference. All the members of the group can be found in the MODELLER `'CHAINS_3.0_40_XN.grp'` file. The `'%ID1'` and `'%ID2'` columns report the percentage sequence

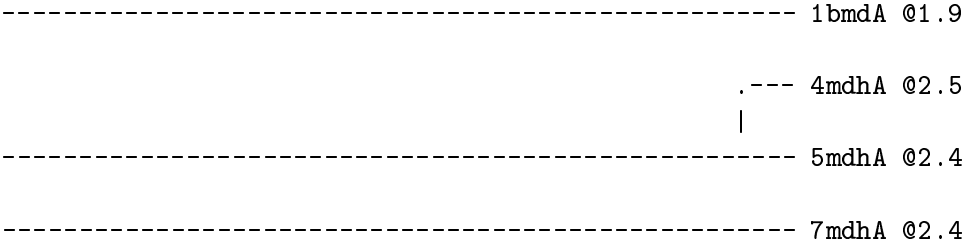
identities between TvLDH and a PDB sequence normalized by their lengths, respectively. In general, a '%ID' value above approximately 25% indicates a potential template unless the alignment is short (*i.e.*, less than 100 residues). A better measure of the significance of the alignment is given by the 'SIGNI' column [72]. A value above 6.0 is generally significant irrespective of the sequence identity and length. In this example, one protein family represented by 1bdmA shows significant similarity with the target sequence, at more than 40% sequence identity. While some other hits are also significant, the differences between 1bdmA and other top scoring hits are so pronounced that we use only the first hit as the template. As expected, 1bdmA is a malate dehydrogenase (from a thermophilic bacterium). Other structures closely related to 1bdmA (and thus not scanned against by **SEQUENCE_SEARCH**) can be extracted from the 'CHAINS_3.0_40_XN.grp' file: 1b8vA, 1bmdA, 1b8uA, 1b8pA, 1bdmA, 1bdmB, 4mdhA, 5mdhA, 7mdhA, 7mdhB, and 7mdhC. All these proteins are malate dehydrogenases. During the project, all of them and other malate and lactate dehydrogenase structures were compared and considered as templates (there were 19 structures in total). However, for the sake of illustration, we will investigate only four of the proteins that are sequentially most similar to the target, 1bmdA, 4mdhA, 5mdhA, and 7mdhA. The following script performs all pairwise comparisons among the selected proteins (file 'compare.top').

```
READ_ALIGNMENT FILE = '$(LIB)/CHAINS_all.seq',;
     ALIGN_CODES = '1bmdA' '4mdhA' '5mdhA' '7mdhA'
MALIGN
MALIGN3D
COMPARE
ID_TABLE
DENDROGRAM
```

The **READ_ALIGNMENT** command reads the protein sequences and information about their PDB files. **MALIGN** calculates their multiple sequence alignment, used as the starting point for the multiple structure alignment. The **MALIGN3D** command performs an iterative least-squares superposition of the four 3D structures. **COMPARE** command compares the structures according to the alignment constructed by **MALIGN3D**. It does not make an alignment, but

it calculates the RMS and DRMS deviations between atomic positions and distances, differences between the mainchain and sidechain dihedral angles, percentage sequence identities, and several other measures. Finally, the **ID_TABLE** command writes a file with pairwise sequence distances that can be used directly as the input to the **DENDROGRAM** command (or the clustering programs in the PHYLIP package [42]). **DENDROGRAM** calculates a clustering tree from the input matrix of pairwise distances, which helps visualizing differences among the template candidates. Excerpts from the log file are shown below (file '`compare.log`').

```
>> Least-squares superposition (FIT)           :       T

   Atom types for superposition/RMS (FIT_ATOMS): CA
   Atom type for position average/variability (DISTANCE_ATOMS[1]): CA

   Position comparison (FIT_ATOMS):

       Cutoff for RMS calculation:     3.5000

       Upper = RMS, Lower = numb equiv positions

           1bmdA   4mdhA   5mdhA   7mdhA
1bmdA      0.000   1.038   0.979   0.992
4mdhA        310   0.000   0.504   1.210
5mdhA        308     329   0.000   1.173
7mdhA        320     306     307   0.000

>> Sequence comparison:

       Diag=numb res, Upper=numb equiv res, Lower = % seq ID

            1bmdA   4mdhA   5mdhA   7mdhA
1bmdA         327     168     168     158
4mdhA          51     333     328     137
5mdhA          51      98     333     138
7mdhA          48      41      41     351
```



The comparison above shows that 5mdhA and 4mdhA are almost identical, both sequentially and structurally. They were solved at similar resolutions, 2.4 and 2.5Å, respectively. However, 4mdhA has a better crystallographic R-factor (16.7 *versus* 20%), eliminating 5mdhA. Inspection of the PDB file for 7mdhA reveals that its crystallographic refinement was based on 1bmdA. In addition, 7mdhA was refined at a lower resolution than 1bmdA (2.4 *versus* 1.9Å), eliminating 7mdhA. These observations leave only 1bmdA and 4mdhA as potential templates. Finally, 4mdhA is selected because of the higher overall sequence similarity to the target sequence.

3.1.3 Aligning TvLDF with the template

A good way of aligning the sequence of TvLDH with the structure of 4mdhA is the **ALIGN2D** command in MODELLER. Although ALIGN2D is based on a dynamic programming algorithm [75], 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 C_α positions that are close in space. As a result, the alignment errors are reduced by approximately one third relative to those that occur with standard sequence alignment techniques. This improvement becomes more important as the similarity between the sequences decreases and the number of gaps increases. In the current example, the template–target similarity is so high that almost any alignment method with reasonable parameters will result in the same alignment. The following MODELLER script aligns the TvLDH sequence in file '`TvLDH.seq`' with the 4mdhA structure in the PDB file '`4mdh.pdb`' (file '`align2d.top`').

```
READ_MODEL FILE = '4mdh.pdb'
SEQUENCE_TO_ALI ALIGN_CODES = '4mdhA'
READ_ALIGNMENT FILE = 'TvLDH.ali', ALIGN_CODES = 'TvLDH', ADD_SEQUENCE = ON
ALIGN2D
WRITE_ALIGNMENT FILE='TvLDH-4mdh.ali', ALIGNMENT_FORMAT = 'PIR'
WRITE_ALIGNMENT FILE='TvLDH-4mdh.pap', ALIGNMENT_FORMAT = 'PAP'
```

In the first line, MODELLER reads the 4mdhA structure file. The **SEQUENCE_TO_ALI** command transfers the sequence to the alignment array and assigns it the name of '`4mdhA`' (ALIGN_CODES). The third line reads the TvLDH sequence from file '`TvLDH.seq`', assigns it the name '`TvLDH`' (ALIGN_CODES) and adds it to the alignment array ('ADD_SEQUENCE = `ON`'). The fourth line executes the **ALIGN2D** command to perform the alignment. Finally, the alignment is written out in two formats, PIR ('`TvLDH-4mdh.ali`') and PAP ('`TvLDH-4mdh.pap`'). The PIR format is used by MODELLER in the subsequent model building stage. The PAP alignment format is easier to inspect visually. Due to the high target–template similarity, there are only a few gaps in the alignment. In the PAP format, all identical positions are marked with a '`*`' (file '`TvLDH-4mdh.pap`').

```
 _aln.pos         10        20        30        40        50        60
4mdhA     GSEPIRVLVTGAAGQIAYSLLYSIGNGSVFGKDQPIILVLLDITPMMGVLDGVLMELQDCALPLLKDV
TvLDH     MSEAAHVLITGAAGQIGYILSHWIASGELYG-DRQVYLHLLDIPPAMNRLTALTMELEDCAFPHLAGF
 _consrvd  **   ** ******* * *   *  *   * *    * **** * *  *    *** *** * *


 _aln.p   70        80        90       100       110       120       130
4mdhA     IATDKEEIAFKDLDVAILVGSMPRRDGMERKDLLKANVKIFKCQGAALDKYAKKSVKVIVVGNPANTN
TvLDH     VATTDPKAAFKDIDCAFLVASMPLKPGQVRADLISSNSVIFKNTGEYLSKWAKPSVKVLVIGNPDNTN
 _consrvd  **     **** * * ** ***   *  * **   *  ***  *  * * ** **** * *** ***


 _aln.pos  140       150       160       170       180       190       200
4mdhA     CLTASKSAPSIPKENFSCLTRLDHNRAKAQIALKLGVTSDDVKNVIIWGNHSSTQYPDVNHAKVKLQA
TvLDH     CEIAMLHAKNLKPENFSSLSMLDQNRAYYEVASKLGVDVKDVHDIIVWGNHGESMVADLTQATFTKEG
 _consrvd *  *   *     **** *  ** ***    * ****   **   * ****      *   *


 _aln.pos    210       220       230       240       250       260       270
4mdhA     KEVGVYEAVKDDSWLKGEFITTVQQRGAAVIKARKLSSAMSAAKAICDHVRDIWFGTPEGEFVSMGII
TvLDH     KTQKVVDVLDHDYVFDTFFKKIGHRAWDILEHRGFTSAASPTKAAIQHMKAWLFGTAPGEVLSMGIPV
 _consrvd *   *      *      *                 * *     **           *


 _aln.pos      280       290       300       310       320       330
4mdhA     SDGNSYGVPDDLLYSFPVTIK-DKTWKIVEGLPINDFSREKMDLTAKELAEEKETAFEFLSSA-
TvLDH     PEGNPYGIKPGVVFSFPCNVDKEGKIHVVEGFKVNDWLREKLDFTEKDLFHEKEIALNHLAQGG
 _consrvd   ** **       ***           ***   **  *** * * * *  *** *   *
```

3.1.4 Model building

Once a target–template alignment is constructed, MODELLER calculates a 3D model of the target completely automaticaly. The following script will generate five similar models of TvLDH based on the 4mdhA template structure and the alignment in file 'TvLDH-4mdh.ali' (file 'model-single.top').

```
INCLUDE
SET ALNFILE = 'TvLDH-4mdh.ali'
SET KNOWNS = '4mdhA'
SET SEQUENCE = 'TvLDH'
SET STARTING_MODEL = 1
SET ENDING_MODEL = 5
CALL ROUTINE = 'model'
```

The first line includes many standard variable and routine definitions. The following five lines set parameter values for the 'model' routine. ALNFILE names the file that contains the target–template alignment in the PIR format. KNOWNS defines the known template structure(s) in ALNFILE ('TvLDH-4mdh.ali'). SEQUENCE defines the name of the target sequence in ALNFILE. STARTING_MODEL and ENDING_MODEL define the number of models that are calculated (their indices will run from 1 to 5). The last line in the file calls the 'model' routine that actually calculates the models. The most important output files are 'model.log', which reports warnings, errors and other useful information including the input restraints used for modeling that remain violated in the final model; and 'TvLDH.B99990001', which contains the model coordinates in the PDB format. The model can be viewed by any program that reads the PDB format, such as MODVIEW (http://guitar.rockefeller.edu/modview/).

3.1.5 Evaluating a model

If several models are calculated for the same target, the "best" model can be selected by picking the model with the lowest value of the MODELLER objective function, which is reported in the second line of the model PDB file. The value of the objective function in MODELLER is not an absolute

measure in the sense that it can only be used to rank models calculated from the same alignment.

Once a final model is selected, there are many ways to assess it (Section 2.5). In this example, ProsaII [45] is used to evaluate the model fold and Procheck [69] is used to check the model's stereochemistry. Before any external evaluation of the model, one should check the log file from the modeling run for runtime errors (`'model.log'`) and restraint violations (see the Modeller manual for details [72]). Both ProsaII and Procheck confirm that a reasonable model was obtained, with a Z-score comparable to that of the template (-10.53 and -12.69 for the model and the template, respectively). However, the ProsaII energy profile indicates an error in the long active site loop between residues 90 and 100 (Figure 4). This loop interacts with region 220-250, that forms the other half of the active site. This latter part is well resolved in the template and probably correctly modeled in the target structure, but due to the unfavourable non-bonded interactions with the 90-100 region, it is also reported to be in error by PROSA. In general, an error indicated by ProsaII is not neccessarily an actual error, especially if it highlights an active site or a protein–protein interface. However, in this case, the same active site loops have a better profile in the template structure, which strengthens the assessment that the model is probably incorrect in the active site region.

3.2 Example 2: Modeling of a protein–ligand complex based on multiple templates and user specified restraints

An important aim of modeling is to contribute to understanding of the function of the modeled protein. Inspection of the 4mdhA template structure revealed that loop 93–100, one of the functionally most important part of the enzyme, is more disordered than the rest of the protein. The long active site loop appears to be flexible in the absence of a ligand and could not be seen well in the diffraction map. The unreliability of the template coordinates and the inability of Modeller to model long insertions is why this loop was poorly modeled in TvLDH, as indicated by ProsaII

(Figure 4). Since we are interested in understanding differences in specificity between two similar proteins, we need to build precise and accurate models. Therefore, we need to search for another template malate dehydrogenase structure, which may have a lower overall sequence similarity to TvLDH, but a better resolved active site loop. The old and new templates can then be used together to get a model of TvLDH. The active site loop tends to be more defined if the structure is solved together with its physiological ligand and a co-factor. The model based on a template with ligands bound is also expected to be more relevant for the purposes of our study of enzymatic specificity, especially if we also build the model with the ligands.

1emd, a malate dehydrogenase from *E. coli* was identified in PDB. While the 1emd sequence shares only 32% sequence identity with TvLDH, the active site loop and its environment are more conserved. The loop in the 1emd structure is well resolved. Moreover, 1emd was solved in the presence of a citrate substrate analog and the NADH cofactor. The new alignment in the PAP format is shown below (file `'TvLDH-4mdh.pap'`).

```
 _aln.pos         10        20        30        40        50        60
1emd_ed   ------------------------------------------------------------
4mdhA     -SEPIRVLVTGAAGQIAYSLLYSIGNGSVFGKDQPIILVLLDITPMMGVLDGVLMELQDCALPLLKDV
TvLDH     MSEAAHVLITGAAGQIGYILSHWIASGELYG-DRQVYLHLLDIPPAMNRLTALTMELEDCAFPHLAGF

 _aln.p   70        80        90       100       110       120       130
1emd_ed   ------------------SAGVRRKPGMDRSDLFNV--------------NAGI--------------
4mdhA     IATDKEEIAFKDLDVAILVGSM--------------PRRDGMERKDLLKANVKIFKCQGAALDKYAKK
TvLDH     VATTDPKAAFKDIDCAFLVASMPLKPGQVRADLISS--------------NSVIFKNTGEYLSKWAKP

 _aln.pos  140       150       160       170       180       190       200
1emd_ed   ------------------------------------------------------------------
4mdhA     SVKVIVVGNPANTNCLTASKSAPSIPKENFSCLTRLDHNRAKAQIALKLGVTSDDVKNVIIWGNHSST
TvLDH     SVKVLVIGNPDNTNCEIAMLHAKNLKPENFSSLSMLDQNRAYYEVASKLGVDVKDVHDIIVWGNHGES

 _aln.pos    210       220       230       240       250       260       270
1emd_ed   ------------------------------------------------------------------
4mdhA     QYPDVNHAKVKLQAKEVGVYEAVKDDSWLKGEFITTVQQRGAAVIKARKLSSAMSAAKAICDHVRDIW
TvLDH     MVADLTQATFTKEGKTQKVVDVLDHD-YVFDTFFKKIGHRAWDILEHRGFTSAASPTKAAIQHMKAWL

 _aln.pos      280       290       300       310       320       330       340
1emd_ed   ------------------------------------------------------------------
4mdhA     FGTPEGEFVSMGIISD-GNSYGVPDDLLYSFPVTIK-DKTWKIVEGLPINDFSREKMDLTAKELAEEK
TvLDH     FGTAPGEVLSMGIPVPEGNPYGIKPGVVFSFPCNVDKEGKIHVVEGFKVNDWLREKLDFTEKDLFHEK

 _aln.pos        350       360       370       380       390       400
1emd_ed   ----------VKNLVQQVAKTCPKACIGIITNPVNTTVAIAAEVLKKAGVYDKNKLFGVTTLDIIRSN
4mdhA     ETAFEFLSSA------------------------------------------------------------
TvLDH     EIALNHLAQ-------------------------------------------------------------

 _aln.p  410       420       430       440       450       460       470
1emd_ed   TFVAELKGKQPGEVEVPVIGGHSGVTILPLLSQVPGVSFTEQEVADLTKRIQNAGTEVVEAKAGGGSA
4mdhA     ------------------------------------------------------------------
TvLDH     ------------------------------------------------------------------

 _aln.pos  480       490       500       510       520       530       540
1emd_ed   TLSMGQAAARFGLSLVRALQGEQGVVECAYVEGDGQYARFFSQPLLLGKNGVEERKSIGTLSAFEQNA
4mdhA     ------------------------------------------------------------------
TvLDH     ------------------------------------------------------------------

 _aln.pos    550       560
1emd_ed   LEGMLDTLKKDIALGQEFVNK/-..
4mdhA     ---------------------/.--
TvLDH     ---------------------/..-
```

The modified alignment refers to an edited 1emd structure (see below), 1emd_ed, as a second

template. The alignment corresponds to a model that is based on 1emd_ed in its active site loop and on 4mdhA in the rest of the fold. Four residues on both sides of the active site loop are aligned with both templates to ensure that the loop has a good orientation relative to the rest of the model.

The modeling script below has several changes with respect to '`model-single.top`'. First, the name of the alignment file assigned to ALNFILE is updated. Next, the variable KNOWNS is redefined to include both templates. Another change is an addition of the '`SET HETATM_IO = ON`' command to allow reading of the non-standard pyruvate and NADH residues from the input PDB files. The script is shown next (file '`model-multiple-hetero.top`').

```
INCLUDE
SET ALNFILE = 'TvLDH-4mdh-1emd_ed.ali'
SET KNOWNS = '4mdhA' '1emd_ed'
SET SEQUENCE = 'TvLDH'
SET STARTING_MODEL = 1
SET ENDING_MODEL = 5
SET HETATM_IO = ON
CALL ROUTINE = 'model'

SUBROUTINE ROUTINE = 'special_restraints'
   ADD_RESTRAINT ATOM_IDS =  'NH1:161' 'O1A:335', ;
       RESTRAINT_PARAMETERS = 2 1 1 22 2 2 0 3.5 0.1
   ADD_RESTRAINT ATOM_IDS =  'NH2:161' 'O1B:335', ;
       RESTRAINT_PARAMETERS = 2 1 1 22 2 2 0 3.5 0.1
   ADD_RESTRAINT ATOM_IDS =  'NE2:186' 'O2:335', ;
       RESTRAINT_PARAMETERS = 2 1 1 22 2 2 0 3.5 0.1
   RETURN
END_SUBROUTINE
```

A ligand can be included in a model in two ways by MODELLER. The first case corresponds to the ligand that is not present in the template structure, but is defined in the MODELLER residue topology library. Such ligands include water molecules, metal ions, nucleotides, heme groups, and many other ligands (see FAQ 18 in the MODELLER manual). This situation is not explored further here. The second case corresponds to the ligand that is already present in the template structure. We can assume either that the ligand interacts similarly with the target and the template, in which

case we can rely on MODELLER to extract and satisfy distance restraints automatically, or that the relative orientation is not necessarily conserved, in which case the user needs to supply restraints on the relative orientation of the ligand and the target (the conformation of the ligand is assumed to be rigid). The two cases are illustrated by the NADH cofactor and pyruvate modeling, respectively. Both NADH and cofactor are indicated by the '.' characters at the end of each sequence in the alignment file above (the '/' character indicates a chain break). In general, the '.' character in MODELLER indicates an arbitrary generic residue called a "block" residue (for details see the MODELLER manual [72]). The 1emd structure file contains a citrate substrate analog. To obtain a model with pyruvate, the physiological substrate of TvLDH, we convert the citrate analog in 1emd into pyruvate by deleting the $-CH(COOH)_2$ group, thus obtaining the 1emd_ed template file. A major advantage of using the '.' characters is that it is not necessary to define the residue topology.

To obtain the restraints on pyruvate, we first superpose the structures of several LDH and MDH enzymes solved with ligands. Such a comparison allows to identify absolutely conserved electrostatic interactions involving catalytic residues Arg 161 and His 186 on one hand, and the oxo groups of the lactate and malate ligands on the other hand. The modeling script can now be expanded by appending a routine that specifies the user defined distance restraints between the conserved atoms of the active site residues and their substrate.

The **ADD_RESTRAINT** command has two arguments. **ATOM_IDS** defines the restrained atoms, by specifying their atom types and the residue numbers as listed in the model coordinate file. RESTRAINT_PARAMETERS defines the restraints, by specifying the mathematical form (*e.g.*, harmonic, cosine, cubic spline), modality, the type of the restrained feature (*e.g.*, distance, angle, dihedral angle), the number of atoms in the restraint, and the restraint parameters. In this case, a harmonic upper bound restraint of 3.5 ± 0.1Å is imposed on the distances between the specified pairs of atoms. A trick is used to prevent MODELLER from automatically calculating distance restraints on the pyruvate–TvLDH complex; the ligand in the 1emd_ed template is moved beyond

the upper bound on the ligand–protein distance restraints (*i.e.*, 10Å).

The new script produces a model with a significantly improved ProsaII profile (Figure 4). The predicted error in the 90-100 active site loop is much less and practically resolved in the loop region 220-250. The overall Z-score is improved from -10.7 to -11.7, which compares well with the template Z-score of -12.7. With this favorable evaluation, we gain confidence in the final model. The model was used for interpreting site-directed mutagenesis experiments aimed at elucidating the determinants of enzyme specificity in this class of enzymes [73].

3.3 Example 3: Modeling the fold and a loop in circularly permuted cyanovirin

Cyanovirin-N (CV-N) was originally isolated from *Nostoc ellipsosporum*. It was identified in a screening effort as a highly potent inhibitor of diverse laboratory adapted strains and clinical isolates of HIV-1, HIV-2 and SIV. Subsequently, the structure of CV-N was solved, first by NMR spectroscopy and later by X-ray crystallography at a resolution of 1.5Å. The two structures are very similar. The CN-V monomer consists of two similar domains with 32% sequence identity to each other. In the crystal structure, the domains are connected by a flexible linker region, forming a dimer by inter-molecular domain swapping.

Recently, work was initiated to solve the monomer structure of a CN-V variant with circularly permuted domains (cpCN-V) [76]. Assuming that the overall structure does not change significantly, the new protein can be modeled by comparative modeling. An initial coarse model is built by using the following alignment file in the PAP format (file `'circ.pap'`).

```
 _aln.pos         10        20        30        40        50        60
2ezm      LGKFSQTCYNSAIQGSVL-TSTCERTNGGYNTSSIDLNSVIENVDGSLKWQPSNFIETCR
cpCN-V    LGKFIETCRNTQLAGSSELAAECKTRAQQFVSTKINLDDHIANIDGTLKWQPSNFSQTCY
          **                                                  ******

 _aln.pos 70        80        90       100
2ezm      NTQLAGSSELAAECKTRAQQFVSTKINLDDHIANIDGTLKYE
cpCN-V    NSAIQGSVL-TSTCERTNGGYNTSSIDLNSVIENVDGSLKYE
                                              **
```

Next, the new linker loop and the short N- and C-termini are refined by *ab initio* loop modeling. The selected segments that are subjected to loop modeling are indicated by stars in the alignment above. The loop modeling script is as follows (file '`loop.top`').

```
INCLUDE
SET SEQUENCE = 'cpCN-V'
SET LOOP_MODEL = 'cpCN-V.pdb'
SET LOOP_STARTING_MODEL = 1
SET LOOP_ENDING_MODEL = 200
CALL ROUTINE = 'loop'

SUBROUTINE ROUTINE = 'select_loop_atoms'
    PICK_ATOMS SELECTION_SEGMENT = '0:' '3:', SELECTION_STATUS = 'initialize'
    PICK_ATOMS SELECTION_SEGMENT = '99:' '100:', SELECTION_STATUS = 'add'
    PICK_ATOMS SELECTION_SEGMENT = '49:' '54:', SELECTION_STATUS = 'add'
    RETURN
END_SUBROUTINE
```

SEQUENCE defines the name of the model. **LOOP_MODEL** defines the name of the input coordinate file containing the cpCN-V model whose loops need to be refined. **LOOP_STARTING_MODEL** and **LOOP_ENDING_MODEL** define how many final loop models are calculated (in this case, 200). The subroutine '`select_loop_atoms`' selects regions of the model for loop modeling. Two arguments are submitted to the **PICK_ATOMS** command. SELECTION_SEGMENT defines the starting and ending residues of the loop. SELECTION_STATUS defines whether or not the program initializes the selection or adds the current loop to the previously defined set of loops. In this case, three loops are selected and optimized simultaneously. The filenames of output models with refined loops have the '`.BL`' extension to distinuish them from the default file

naming convention of the regular models (`'.B'`). For instance, the first generated loop model file is `'cpCN-V.BL00010001'`.

Although the linker segment is only six residues long, it is not known whether or not some of the preceding and subsequent residues undergo conformational changes in the new construct. To investigate this question, we gradually extended the length of the modeled linker region from 6 to 12 residues. For this purpose, one needs to modify only the selection routine in the script above.

The model with the lowest energy score of the 200 generated models was selected for each linker length from 6 to 12 residues. The superposition of the best models of varying length showed a dominant cluster of conformations, indicating that the modeling of the linker region is not limited by conformational changes in the immediately preceding or subsequent parts of the sequence (Figure 5). The final comparative model with the optimized linker and terminal segments was used to refine the structure of cpCN-V against NMR dipolar coupling data. A good agreement between the experimental values and those calculated from the model confirmed that the fold of cpCN-V is similar to that of the wild type and that the model may facilitate characterization of the structure and dynamics of cpCV-N [76].

Acknowledgments

We are grateful to all the members of our research group for many discussions about comparative protein structure modeling. AF was a Burroughs Wellcome Fund Postdoctoral Fellow and is a Charles Revson Foundation Postdoctoral Fellow. AS is an Irma T. Hirschl Trust Career Scientist. Research was supported by NIH/GM 54762, Merck Genome Research Award (AS), and Mathers Foundation. This review is based on [6, 7, 77].

MODELLER is available freely to academic users at `http://guitar.rockefeller.edu/modeller/modeller.html`. It runs on many UNIX systems, including PCs running LINUX. All the sample files shown in this review are available at `http://guitar.rockefeller.edu/modeller/methenz/`. MODELLER, with a graphical interface, is also available as part of QUANTA, INSIGHTII and GENE-EXPLORER (Accelrys Inc., San Diego, e-mail: `dje@accelrys.com`).

References

[1] W. J. Browne, A. C. T. North, D. C. Phillips, K. Brew, T. C. Vanaman, and R. C. Hill, J. Mol. Biol., 42, 65–86, 1969.

[2] T. L. Blundell, M. J. E. Sternberg, B. L. Sibanda, and J. M. Thornton, Nature, 326, 347–352, 1987.

[3] J. Bajorath, R. Stenkamp, and A. Aruffo, Protein Sci., 2, 1798–1810, 1994.

[4] M. S. Johnson, N. Srinivasan, R. Sowdhamini, and T. L. Blundell, CRC Crit. Rev. Biochem. Mol. Biol., 29, 1–68, 1994.

[5] R. Sánchez and A. Šali, Curr. Opin. Struct. Biol., 7, 206–214, 1997.

[6] M. A. Martí-Renom, A. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali, Ann. Rev. Biophys. Biomolec. Struct., 29, 291–325, 2000.

[7] A. Fiser, R. Sánchez, F. Melo, and A. Šali. Comparative protein structure modeling. In M. Watanabe, B. Roux, A. MacKerell, and O. Becker, editors, Computational Biochemistry and Biophysics, in press, pages 275–312. Marcel Dekker, 2000.

[8] D. Baker, Nature, 405, 39–42, 2000.

[9] A. M. Lesk and C. Chothia, J. Mol. Biol., 136, 225–270, 1980.

[10] R. Sánchez, U. Pieper, F. Melo, N. Eswar, M.A. Martí-Renom, M.S. Madhusudhan, N. Mirković, and A. Šali, Nat. Struct. Biol., 7, 986–990, 2000.

[11] A. J. Jennings and M. J. Sternberg, Prot. Eng., 14, 227–231, 2001.

[12] D. A. Benson, M. S. Boguski, D. J. Lipman, J. Ostell, B. F. F. Ouellette, B. A. Rapp, and D. L. Wheeler, Nucl. Acids Res., 27, 12–17, 1999.

[13] A. Bairoch and R. Apweiler, Nucl. Acids Res., 27, 49–54, 1999.

[14] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, Nucleic Acids Res., 28, 235–242, 2000.

[15] C. Chothia, Nature, 360, 543–544, 1992.

[16] T. J. P. Hubbard, B. Ailey, S. E. Brenner, A. G. Murzin, and C. Chothia, Nucl. Acids Res., 27, 254–256, 1999.

[17] L. Holm and C. Sander, Nucl. Acids Res., 27, 244–247, 1999.

[18] J.E. Bray, A.E. Todd, F.M. Pearl, J.M. Thornton, and C.A. Orengo, Protein Eng, 13, 153–65, 2000.

[19] L. Holm and C. Sander, Science, 273, 595–602, 1996.

[20] S. K. Burley, S. C. Almo, J. B. Bonanno, , M. Capel, M. R. Chance, T. Gaasterland, D. Lin, A. Šali, F. W. Studier, and S. Swaminathan, Nat. Genet., 23, 151–157, 1999.

[21] Nat. Str. Biol. Suppl., 2000.

[22] A. Šali and T. L. Blundell, J. Mol. Biol., 234, 779–815, 1993.

[23] A. Šali and J.P Overington, Protein Sci., 3, 1582–1596, 1994.

[24] A. Fiser, R. K. G. Do, and A. Šali, Protein Science, 9, 1753–1773, 2000.

[25] S. F. Altschul, M. S. Boguski, W. Gish, and J. C. Wootton, Nature Genetics, 6, 119–129, 1994.

[26] W. R. Pearson, Methods Enzymol., 266, 227–258, 1996.

[27] G.D. Schuler, Methods Biochem. Anal., 39, 145–171, 1998.

[28] G. J. Barton. Protein sequence alignment and database scanning. In M. J. E. Sternberg, editor, Protein Structure Prediction: A Practical Approach. IRL Press at Oxford University Press, 1998.

[29] M. Levitt and M. Gerstein, Proc. Natl. Acad. Sci. USA, 95, 5913–5920, 1998.

[30] M. Gribskov, Meth. Mol. Biol., 25, 247–266, 1994.

[31] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler, J. Mol. Biol., 235, 1501–1531, 1994.

[32] S. R. Eddy, Curr. Opin. Struct. Biol., 6, 361–365, 1996.

[33] K. Karplus, C. Barrett, and R. Hughey, Bioinformatics, 14, 846–856, 1998.

[34] J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and Chothia C., J. Mol. Biol., 284, 201–210, 1998.

[35] J. U. Bowie, R. Lüthy, and D. Eisenberg, Science, 253, 164–170, 1991.

[36] D. T. Jones, W. R. Taylor, and J. M. Thornton, Nature, 358, 86–89, 1992.

[37] A. Godzik, A. Kolinski, and J. Skolnick, J. Mol. Biol., 227, 227–238, 1992.

[38] M. J. Sippl and H. Flöckner, Structure, 4, 15–19, 1996.

[39] A. E. Torda, Curr. Opin. Struct. Biol., 7, 200–205, 1997.

[40] H. Lu and J. Skolnick, Proteins, 44, 223–232, 2001.

[41] R. L. Dunbrack Jr., D. L. Gerloff, M. Bower, X. Chen, O. Lichtarge, and F. E. Cohen, Folding & Design, 2, R27–R42, 1997.

[42] J. Felsenstein, Evolution, 39, 783–791, 1985.

[43] A. Šali, L. Potterton, F. Yuan, H. van Vlijmen, and M. Karplus, Proteins, 23, 318–326, 1995.

[44] R. Sánchez and A. Šali, Proteins, Suppl. 1, 50–58, 1997.

[45] M. J. Sippl, Proteins, 17, 355–362, 1993.

[46] G. Wu, H. G. Morrison, A. Fiser, A. G. McArthur, A. Šali, M. L. Sogin, and M. Müller, Mol. Biol. Evol., 17, 1156–1163, 2000.

[47] R. Sánchez and A. Šali, Proc. Natl. Acad. Sci. USA, 95, 13597–13602, 1998.

[48] TG. Dewey, J Comput Biol, 8, 177–90, 2001.

[49] J. Shi, T. L. Blundell, and Mizuguchi K., J. Mol. Biol., 310, 243–257, 2001.

[50] J.D. Blake and F.E. Cohen, J. Mol. Biol., 307, 721–35, 2001.

[51] L. Jaroszewski, L. Rychlewski, and A. Godzik, Protein Sci, 9, 1487–96, 2000.

[52] J.M. Sauder, J.W. Arthur, and R.L. Dunbrack, Proteins, 40, 6–22, 2000.

[53] J. Greer, J. Mol. Biol., 153, 1027–1042, 1981.

[54] T. L. Blundell, B. L. Sibanda, M. J. E. Sternberg, and J. M. Thornton, Nature, 326, 347–352, 1987.

[55] T. H. Jones and S. Thirup, EMBO J., 5, 819–822, 1986.

[56] R. Unger, D. Harel, S. Wherland, and J. L. Sussman, Proteins, 5, 355–373, 1989.

[57] M. Claessens, E. V. Cutsem, I. Lasters, and S. Wodak, Protein Eng., 4, 335–345, 1989.

[58] M. Levitt, J. Mol. Biol., 226, 507–533, 1992.

[59] T. F. Havel and M. E. Snow, J. Mol. Biol., 217, 1–7, 1991.

[60] S. Srinivasan, C. J. March, and S. Sudarsanam, Protein Sci., 2, 227–289, 1993.

[61] S. M. Brocklehurst and R. N. Perham, Protein Sci., 2, 626–639, 1993.

[62] A. Aszódi and W. R. Taylor, Folding and Design, 1, 325–334, 1996.

[63] A. D. MacKerell, Jr., D. Bashford, M. Bellott, R.L. Dunbrack Jr., J.D. Evanseck, M.J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F.T.K. Lau, C. Mattos, S. Michnick, T. Ngo, D.T. Nguyen, B. Prodhom, W.E. Reiher, III, M. Roux, B.and Schlenkrich, J.C. Smith, J. Stote, R.and Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, J. Phys. Chem. B, 102, 3586–3616, 1998.

[64] A. Kolinski, M. R. Betancourt, D. Kihara, P. Rotkiewicz, and J. Skolnick, Proteins, 44, 133–149, 2001.

[65] K. Fidelis, P. S. Stern, D. Bacon, and J. Moult, Protein Eng., 7, 953–960, 1994.

[66] M. J. Sippl, J. Mol. Biol., 213, 859–883, 1990.

[67] B. Cheng, A. Nayeem, and H. A. Scheraga, J. Comp. Chem., 17, 1453–1480, 1996.

[68] R. Lüthy, J. U. Bowie, and D. Eisenberg, Nature, 356, 83–85, 1992.

[69] R. A. Laskowski, M. W. McArthur, D. S. Moss, and J. M. Thornton, J. Appl. Cryst., 26, 283–291, 1993.

[70] R.W.W Hooft, G. Vriend, C. Sander, and E.E. Abola, Nature, 381, 272, 1996.

[71] B. Guenther, R. Onrust, A. Šali, M. O'Donnell, and J. Kuriyan, Cell, 91, 335–345, 1997.

[72] A. Šali, R. Sánchez, A. Y. Badretdinov, A. Fiser, F. Melo, J. P. Overington, E. Feyfant, and M. A. Martí-Renom. MODELLER, A Protein Structure Modeling Program, Release 6. URL `http://guitar.rockefeller.edu/`, 2000.

[73] G. Wu, A. Fiser, B. ter Kuile, A. Šali, and M. Müller, Proc. Natl. Acad. Sci. USA, 96, 6285–6290, 1999.

[74] W.C. Barker, J.S. Garavelli, D.H. Haft, L.T. Hunt, C.R. Marzec, B.C. Orcutt, G.Y. Srinivasarao, L.S.L. Yeh, R.S. Ledley, H.W. Mewes, F. Pfeiffer, and A. Tsugita, Nucl. Acids Res., 26, 27–32, 1998.

[75] S. B. Needleman and C. D. Wunsch, J. Mol. Biol., 48, 443–453, 1970.

[76] L. G. Barrientos, R. Campos-Olivas, J. M. Louis, A. Fiser, A. Sali, and A. M. Gronenborn, J. Biomol. NMR, 19, 289–290, 2001.

[77] R. Sánchez and A. Šali. Comparative protein structure modeling: Introduction and practical examples with MODELLER. In D. M. Webster, editor, Protein Structure Prediction: Methods and Protocols, pages 97–129. Humana Press, 2000.

Designing (site-directed) mutants to test hypotheses about function
Identifying active and binding sites
Searching for ligands of a given binding site
Designing and improving ligands of a given binding site
Modeling substrate specificity
Predicting antigenic epitopes
Protein–protein docking simulations
Inferring function from calculated electrostatic potential around the protein
Molecular replacement in X-ray structure refinement
Refining models against NMR dipolar coupling data
Testing a given sequence – structure alignment
Rationalizing known experimental observations
Planning new experiments

Table 1: Common uses of comparative protein structure models. A list of our papers using MODELLER to address practical problems in collaboration with experimentalists can be obtained at URL `http://guitar.rockefeller.edu/publications/ref/ref.html`.

Databases	
NCBI	www.ncbi.nlm.nih.gov/
PDB	www.rcsb.org/
MSD	www.rcsb.org/databases.html
CATH	www.biochem.ucl.ac.uk/bsm/cath/
TrEMBL	srs.ebi.ac.uk/
Scop	scop.mrc-lmb.cam.ac.uk/scop/
Presage	presage.stanford.edu
ModBase	guitar.rockefeller.edu/modbase/
GeneCensus	bioinfo.mbb.yale.edu/genome
GeneBank	www.ncbi.nlm.nih.gov/Genbank/GenbankSearch.html
PSI	www.structuralgenomics.org
Template search, fold assignment	
PDB-Blast	bioinformatics.burnham-inst.orgpdb_blast
BLAST	www.ncbi.nlm.nih.gov/BLAST/
FastA	www.dna.affrc.go.jp/htdocs/Blast/fasta.html
DALI	www2.ebi.ac.uk/dali/
PhD, TOPITS	www.embl-heidelberg.de/predictprotein/predictprotein.html
THREADER	insulin.brunel.ac.uk/
123D	genomic.sanger.ac.uk/123D/run123D.html
UCLA-DOE	www.doe-mbi.ucla.edu/people/frsvr/frsvr.html
PROFIT	lore.came.sbg.ac.at/
MATCHMAKER	www.tripos.com/software/mm.html
3D-PSSM	www.bmm.icnet.uk/ 3dpssm/html/ffrecog.html
BIOINGBGU	www.cs.bgu.ac.il/ bioinbgu/
FUGUE	www-cryst.bioc.cam.ac.uk/ fugue
LOOPP	ser-loopp.tc.cornell.edu/loopp.html
FASS	bioinformatics.burnham-inst.org/FFAS/index.html
SAM-T99/T98	www.cse.ucsc.edu/research/compbio/sam.html

Table 2: Web sites useful for comparative modeling.

Comparative modeling	
3D-JIGSAW	www.bmm.icnet.uk/servers/3djigsaw/
CPH-Models	www.cbs.dtu.dk/services/CPHmodels/
COMPOSER	www-cryst.bioc.cam.ac.uk/
FAMS	physchem.pharm.kitasato-u.ac.jp/FAMS/fams.html
MODELLER	guitar.rockefeller.edu/modeller/modeller.html
PrISM	honiglab.cpmc.columbia.edu/
SWISS-MODEL	www.expasy.ch/swissmod/SWISS-MODEL.html
SDSC1	cl.sdsc.edu/hm.html
WHAT IF	www.cmbi.kun.nl/bioinf/predictprotein/
ICM	www.molsoft.com/
SCWRL	www.fccc.edu/research/labs/dunbrack/scwrl/
InsightII	www.accelrys.com
SYBYL	www.tripos.com
Model evaluation	
PROCHECK	www.biochem.ucl.ac.uk/~roman/procheck/procheck.html
WHATCHECK	www.cmbi.kun.nl/swift/whatcheck/
ProsaII	www.came.sbg.ac.at
BIOTECH	biotech.embl-ebi.ac.uk:8400/
VERIFY3D	www.doe-mbi.ucla.edu/Services/Verify_3D/
ERRAT	www.doe-mbi.ucla.edu/Services/Errat.html
AQUA	urchin.bmrb.wisc.edu/ jurgen/Aqua/server/
SQUID	www.yorvic.york.ac.uk/~oldfield/squid
PROVE	www.ucmb.ulb.ac.be/UCMB/PROVE/

Table 2 continued.

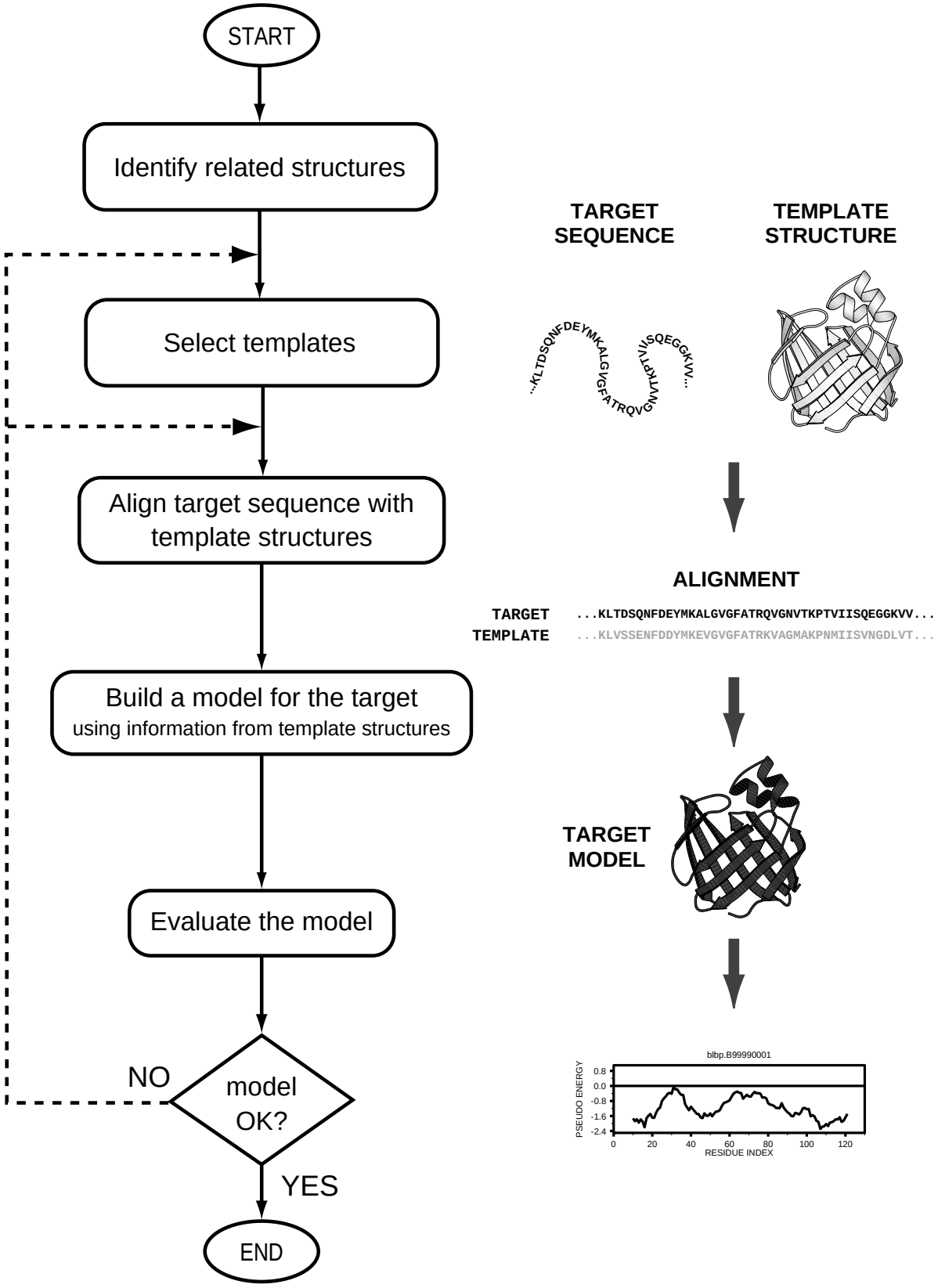


Figure 1: Steps in comparative protein structure modeling. See text for description of each step.

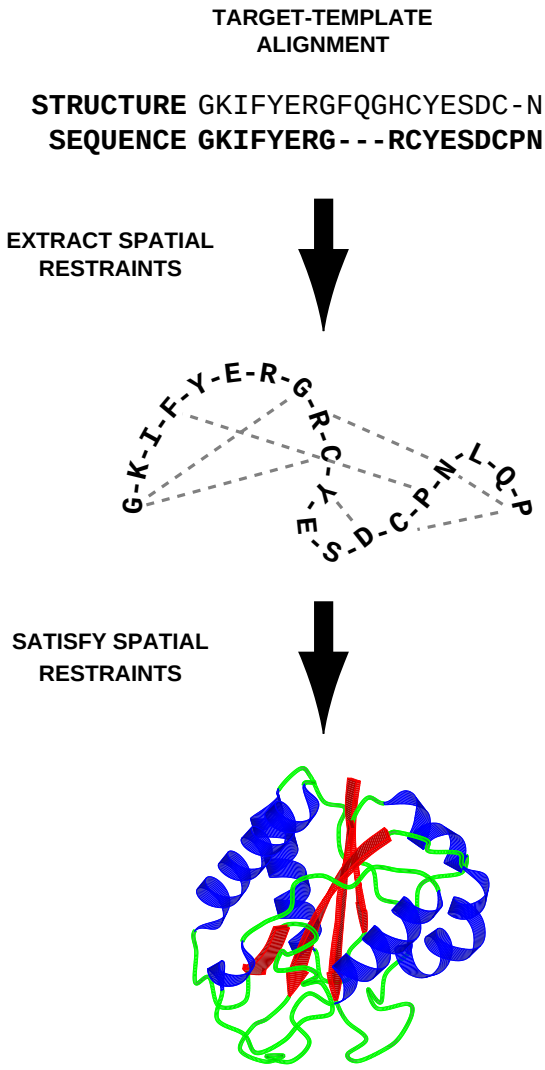


Figure 2: Comparative model building by program MODELLER. First, homology-derived spatial restraints on many atom-atom distances and dihedral angles are extracted from the template structure(s). The alignment is used to determine equivalent residues between the target and the template. The homology-derived and stereochemical restraints are combined into an objective function. Finally, the model of the target is optimized until a model that best satisfies the spatial restraints is obtained. This procedure is similar to the one used in structure determination by NMR spectroscopy.

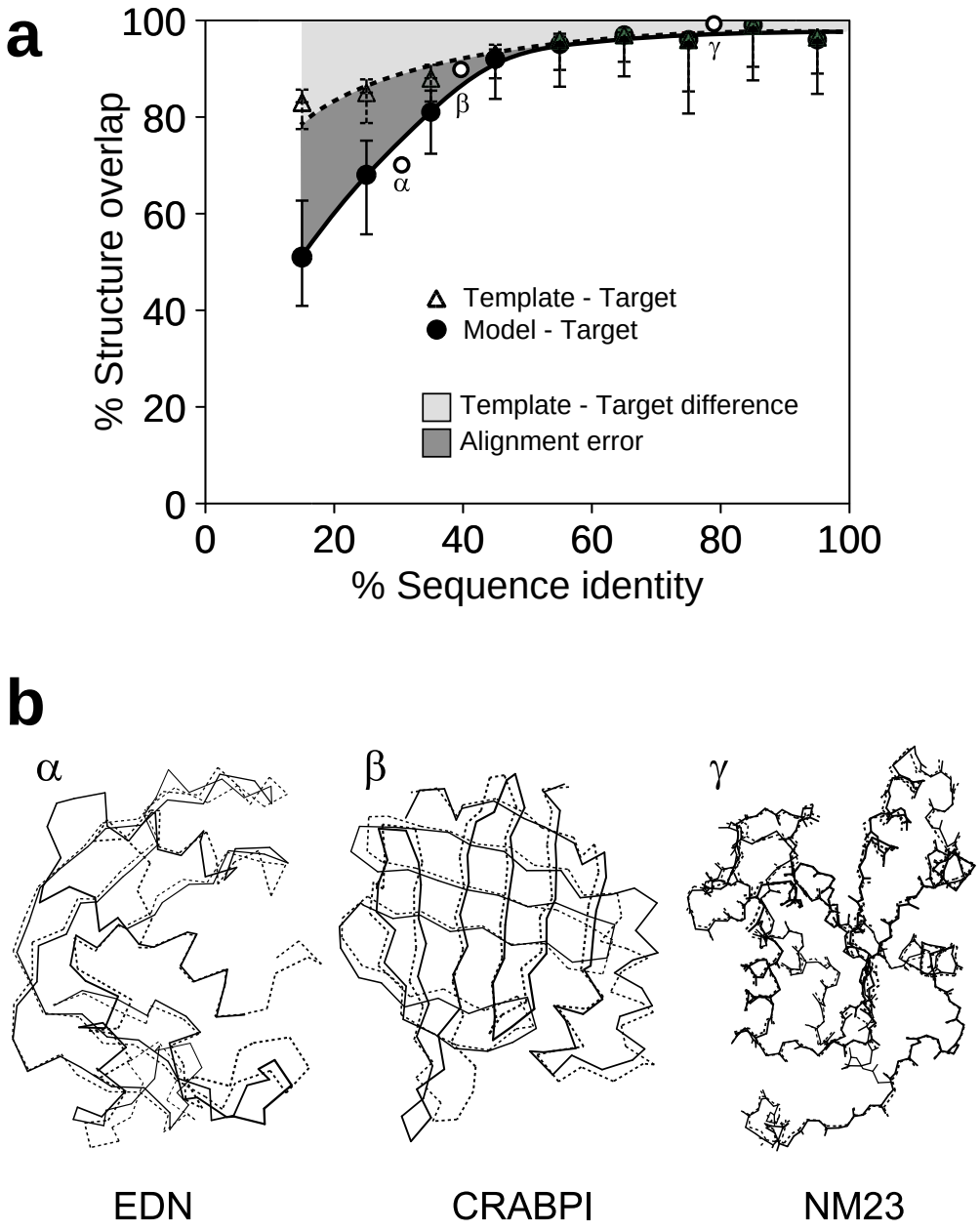


Figure 3: Average model accuracy as a function of sequence identity. As the sequence identity between the target sequence and the template structure decreases, the average structural similarity between the template and the target also decreases (dotted line, open circles). (continued on the next page)

(Figure 3: continued from the previous page)

Structural overlap is defined as the fraction of equivalent C_α atoms. For the comparison of the model with the actual structure (filled circles), two C_α atoms were considered equivalent if they belonged to the same residue and were within 3.5Å of each other after least-squares superposition of all C_α atoms by the ALIGN3D command in MODELLER. For comparison of the template structure with the actual target structure (open circles), two C_α atoms were considered equivalent if they were within 3.5Å of each other after alignment and rigid-body superposition. At high sequence identities, the models are close to the templates and therefore also close to the experimental target structure (solid line, filled circles). At low sequence identities, errors in the target–template alignment become more frequent and the structural similarity of the model with the experimental target structure falls below the target–template structural similarity. The difference between the model and the actual target structure is a combination of the target–template differences (light area) and the alignment errors (dark area). The figure was constructed by calculating 3993 comparative models based on single templates of varying similarity to the targets. All targets had known (experimentally determined) structures and therefore the comparison of the models and templates with the experimental structures was possible [47]. The top part of the figure shows three models (solid line) compared with their corresponding experimental structures (dotted line). The models were calculated with MODELLER in a completely automated fashion before the experimental structures were available [43]. The arrows indicate the target–template similarity in each case.

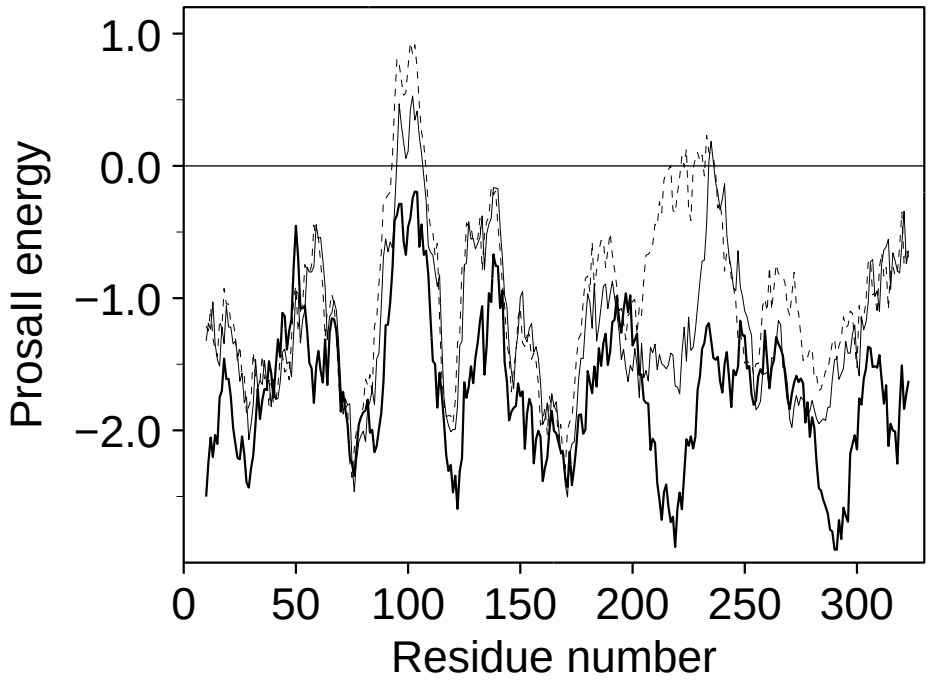


Figure 4: ProsaII [45] energy profile for the raw TvLDH model (dashed line), refined TvLDH model (thin line), and the 4mdhA template structure (heavy line) (Examples 1 and 2). The extended peak above the zero line in the region 90–100 and 220–250 of the raw model highlights a possible error in the raw model, significantly improved in the refined model.

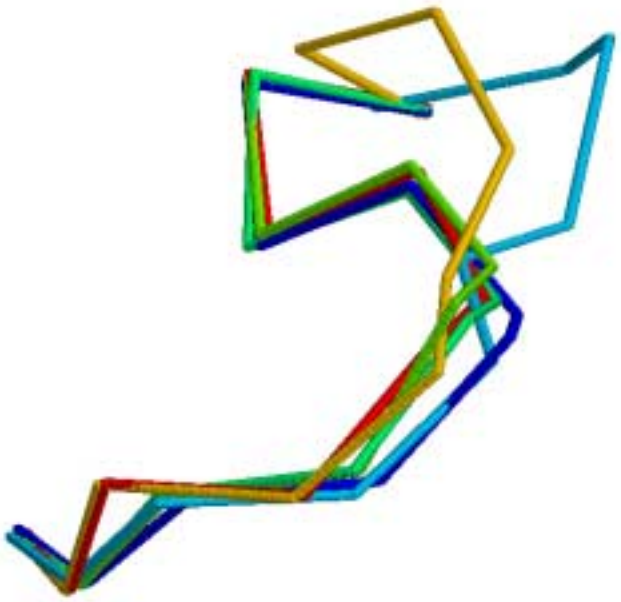


Figure 5: Superposition of models for six linker segments with lengths from 6 to 9 residues. Towards the C-terminus of the loop, a larger structural variation can be observed, but the dominant conformation is well defined by a cluster of four loops.