Published in final edited form as:
Proteins. 2009 September ; 76(4): 930–945. doi:10.1002/prot.22401.

Building and assessing atomic models of proteins from structural templates: Learning and benchmarks

Brinda Kizhakke Vallat[1], **Jaroslaw Pillardy**[2], **Peter Májek**[3], **Jaroslaw Meller**[4,5], **Thomas Blom**[1], **BaoQiang Cao**[1], and **Ron Elber**[1]

[1] Department of Chemistry and Biochemistry, Institute of Computational Engineering and Sciences, University of Texas at Austin, 1 University Station, ICES C0200, Austin TX 78712

[2] Computational Biology Service Unit, Core Laboratories Center and Center for Advanced Computing, Cornell University, Ithaca, New York 14853

[3] Department of Computer Science, Cornell University, Ithaca, New York, 14853

[4] Division of Biomedical Informatics, Children's Hospital Research Foundation, 3333 Burnet Avenue, Cincinnati, Ohio 45229

[5] Departments of Environmental Health and Biomedical Engineering, University of Cincinnati, College of Medicine, 231 Albert Sabin way, Ohio 45267

Abstract

One approach to predict a protein fold from a sequence (a target) is based on structures of related proteins that are used as templates. We present an algorithm that examines a set of candidates for templates, builds from each of the templates an atomically detailed model, and ranks the models. The algorithm performs a hierarchical selection of the best model using a diverse set of signals. After a quick and suboptimal screening of template candidates from the protein data bank, the current method fine-tunes the selection to a few models. More detailed signals test the compatibility of the sequence and the proposed structures, and are merged to give a global fitness measure using linear programming. This algorithm is a component of the prediction server LOOPP (http://www.loopp.org). Large scale training and tests sets were designed and are presented. Recent results of the LOOPP server in CASP8 are discussed.

Keywords

homology modeling; mathematical programming; feature selection; structure determination

I. Introduction

Homology modeling of protein structures is usually divided into three steps. In the first step structural templates are identified from a set of experimentally determined protein structures (the protein data bank PDB [1]). In the second step, an alignment between the sequence of the target with each of the templates is obtained. Based on the alignment, atomically detailed structures are constructed from the templates. The atomically detailed models are finally assessed and ranked. The process of template selection can be difficult. Therefore we divide the template selection process into two sequential steps: (i) template enrichment (Phase 1) and

Full contact info: Ron Elber, Institute of Computational Engineering and Sciences, University of Texas at Austin, 1 University Station, ICES C0200, Austin TX 78712, E-mail: ron@ices.utexas.edu, phone: 512-232-5415, fax: 512-471-8694.

(ii) template focusing and model building (Phase 2). Empirically, we found that to begin with about 1.7% of the target-template pairs are true hits while after Phase 1 enrichment the percentage of true hits increases to 18%.

The following definitions are used throughout the manuscript: a “hit” is a prediction by the algorithm that the proposed match of a template and a target is likely to be successful. A true hit means that the prediction of the algorithm is correct. It is frequently referred to as a T pair or a T match. A false hit is an incorrect template-target match of the algorithm. It is also called a D pair (D for decoy).

Earlier, we presented a paper on a mathematical programming based method for enrichment of suitable templates for target proteins [2]. The present work follows the previous paper and constitutes the next step of the LOOPP server (http://www.loopp.org) for protein structure prediction. Tentative hits identified in first step (Phase 1) are forwarded to Phase 2 where atomically detailed models are built with the program Modeller [3] based on the templates determined during Phase 1 and the alignments of SSALN [4]. The models are assessed using a new learning and scoring algorithm described and discussed in the present manuscript, which constitutes the Phase 2 of the LOOPP server. Phase 2 typically provides a final list of five to twenty top structural candidates to the sequence of the target.

From the perspective of finding the best model the division into Phases is not optimal. The enrichment step may miss some true hits (structural templates that provide good atomic models to the template) and not include them in the subset forwarded to Phase 2. These misses, even if detectable by the filters of Phase 2, obviously remain undetected. Therefore the current LOOPP procedure is less sensitive than an alternative implementation that examines all the PDB structures with the best measures we have at hand. We discuss below the reasons that led to the present computational model of LOOPP.

At the core of the algorithms for Phase 1 and Phase 2 one finds similarity measures that we use to test the fitness of the sequence of the target to the sequence and structure of a template. As discussed in reference [2] the different similarity measures are learned with mathematical programming and are made into scores that rank the pairs of target and templates. The algorithm of Phase 1 uses only a fraction of the similarity measures that are available to us. Not using all of the measures results in a suboptimal performance and less accurate ranking of some of the pairs compared to the ranking of Phase 2. The reason of not using Phase 2 to begin with is computational cost. Some of the similarity measures that are used effectively in Phase 2 are expensive to compute. Phase 1 examines a representative set of the whole Protein Data Bank (PDB) [1] and the large number of comparisons makes it necessary to avoid some of the expensive similarity measures used in Phase 2.

For example, consider the comparison of two sequences. Let the raw optimal score between target i and template j be T_{ij}. The Z score (Z_{ij}) is defined as $Z_{ij}=(T_{ij}-\langle T\rangle)/\sqrt{\langle T^2\rangle-\langle T\rangle^2}$. The brackets $\langle\ldots\rangle$ denote an average over optimal alignments of the template sequence j and randomly shuffled sequences with the same amino acid composition as the target i. To obtain meaningful averages hundreds to thousands alignments are required, making the Z score calculation more expensive than a single alignment by two to three orders of magnitude.

Therefore, despite the observation that the Z score is significantly more sensitive and specific we did not use it in Phase 1. Phase 1 ranking is based on raw scores only (and BLAST statistical evaluation when possible) that are evaluated for 13,875 proteins in the database. Of course a score of sequence alignment is not the only similarity measure that we use to select the candidates of Phase 1. For example, threading and alignment against secondary structure were used as well. (Check the appendix of reference [2] for a complete list of similarity measures

that we used in Phase 1). We then make the assumption that Phase 1 [2] is sufficiently accurate to capture the true hits in the top 200 (from a total of 13,875 candidates). If Phase 1 ranking was perfect (in the sense that the template providing the best structural model is ranked number 1), then only one template is required for further model building. However, it is not. Another complication is that Phase 1 depends on the quality of future steps, such as the alignment (which we perform with SSALN [4]) and the construction of an atomically detailed model (which we do with Modeller [3]). It is likely that modeling of the structures with other programs (e.g., different alignment algorithms or different built up of loops and side chains) will impact the learning.

We strictly differentiate between learning and testing of the prediction model (see section II). We call the set of proteins we learn Learning Set (LS) and use it to optimize the parameters and the functional form of the computational model. The set TS1 includes the proteins of our most comprehensive test case that is built completely independently from the LS. To account for some of the inaccuracies of Phase 1 ranking the number of structures that we forward to Phase 2 is 200. In our learning and test cases of Phase 1 we miss 157992 of the 418037 (LS) and 35759 of the 91449 true hits (TS1). This number may seem highly significant, however, by the end of the day we wish to obtain one good model per protein. We care less about having 100 good models for a particular target sequence. The number of proteins that lost all of their templates is small and stands on 811 out of 12689 proteins (LS) and 198 out of 3802 proteins (TS1). These are fractions of 0.064 (LS) and 0.052 (TS1) from the total number of proteins we have considered. The comparable loss for the learning and test cases is reassuring from the perspective of over-learning and we expect it to be similar for future predictions.

Phase 2, which is discussed in the present manuscript, deals with a much smaller number of candidates for true hits. The limited number of candidates allows for full construction of atomically detailed models for each candidate and the use of comprehensive measures of model accuracy in a calculation feasible on a typical cluster. On a cluster of 20 CPUs a structure prediction of a protein of length 200 amino acids requires 3–5 hours. The time is significantly longer for longer proteins (about 11–15 hours for 500 amino acids), however even this calculation is accessible with moderate computational resources.

The rest of the text is divided between a detailed description of the data sets that were used for training and testing, description of the learned model, and detailed analysis of the performance of the model on various tests. We finally discuss the performance of LOOPP during CASP8.

II. Training, testing, and data sets

II. 1 The learning set

The learning set (LS) follows from the one used in Phase 1 [2]. It constitutes an extensive dataset of proteins selected from the available folds of the Protein Data Bank (PDB) [1] and of the domains of SCOP [5] as of 6/28/2005. In the initial selection, proteins from the PDB with less than 70 percent sequence identity to other members of the set were kept, providing a total of 9,513 single peptide chains. This is supplemented with representative folds from the complete SCOP hierarchy [5] giving a total of 13,825 targets and templates in the learning set, which we call LS. From a total number of 71,824,926 pairs, about 1,187,173 pairs were short listed as probable true templates from the Phase 1 prediction model and forwarded to Phase 2. A complete list of the selected structures is available in http://www.loopp.org/ls2pairs.txt

In Phase 2, the all-atom models are generated for the targets based on all the templates identified in Phase 1. This is done using Modeller [3], which generates an atomically detailed structure of the target based on the fold of the template and an alignment of the target with the template. The alignments were generated by the algorithm SSALN [4] implemented in LOOPP. Modeller

is a widely used resource in the field, and hence is a component in our structure prediction system LOOPP http://www.loopp.org. The learning of scores for optimal alignment in LOOPP is documented elsewhere [4,6] and is beyond the scope of this paper. Here we focus on the identification of the best templates and models for a given target.

The true templates (T) and the decoys (D) from these pairs are identified based on the similarity of the model (built from the template) to the native. Since the accuracy of a predicted model is based on how close it is to the target structure, we define an acceptable model as one within 6.5 Å all-atom RMS distance from the native structure. For each target, we identify the true models using this criterion and define the remaining ones as decoys (D). For targets which do not have any true model using this criterion, we relax the RMS distance cutoff to 7.0 Å to identify the true templates and decoys. We thus obtain 209,090 true hits (T) and 978,083 decoys (D) for a total 12,527 proteins in the LS. Some of these proteins have both T and D representatives, whereas some others have either T or D. 11,694 proteins have one or more T pairs and 11,093 proteins have one or more D pairs as classified above.

Figure 1 shows the histogram of the probability of observing TM score as a function of the TM score for the T pairs. TM align is a structural alignment algorithm that was developed in Skolnick's Laboratory and is used extensively in our studies [7]. Along with the alignment, it also provides a numeric score between 0 and 1, which indicates the degree of structural similarity between the two structures. The higher the score, the closer the structures and a TM score of 0.5 and above is considered significant [7]. We see from Figure 1 that most of the T pairs have TM score of 0.5 or more. Of a total of 209,090 T pairs, about 11,500 pairs have TM scores less than 0.5. Although these are classified as T based on our RMSD criterion, their TM scores indicate that these pairs might not have significant structural similarity.

II. 2 Test sets

Similar to the LS, the test sets used here also follow from Phase 1 and are designed to be independent of LS. We have generated two test cases: TS1 and TS2. In the construction of TS1 we examine all new PDB entries from dates 6/28/2005 to 6/13/2006. Hence, the structures collected did not overlap with the training set. Using the same screening procedure as in LS, we get 4,183 proteins in TS1. Of the 22,096,370 pairs, 310,031 pairs were short listed as probably true hits in Phase 1 and forwarded to Phase 2. These consist of 3,779 individual proteins. The Phase 2 classification procedure yields 39,364 Ts and 270,667 Ds.

The second test set (TS2) is based on CASP7 targets (http://predictioncenter.org/casp7/). CASP is a community wide experiment for critical assessment of methods for protein structure prediction [8,9], where protein sequences with pre-determined but undisclosed structures are provided as targets for prediction and models predicted by various groups are assessed based on their similarity with the native structure. So, our second test set is from the previous experiment of CASP, CASP 7. The CASP7 experiment was conducted from May to July 2006 and therefore our learning set did not overlap with structures from CASP7. We had 82 proteins with 702,828 pairs to begin with in Phase 1. The Phase 1 prediction model forwarded 3451 hits consisting of 55 individual proteins to Phase 2. Using the same criterion as in LS and TS1, we obtain 577 T and 2874 D pairs in the CASP7 based TS2.

In addition to TS1 and TS2 we report preliminary results of LOOPP server on CASP8.

III. A system for identification of best models: Characteristics, Features, Similarity Measures, and Scores

III. 1 The scores

The similarity measures used here are similar to those used in Phase 1 [2], except that we allow the use of more expensive measures that add significantly to the sensitivity of the algorithm at sizable computational cost. Assessing a model in Phase 2 typically requires 51 seconds compared to 0.4 seconds for a pair assessment in Phase 1 (the estimate was for a protein of 189 amino acids). When we compare a target to a template, we have the detailed three-dimensional description of the template, whereas we only have sequence-related information for the target. We built a set of characteristics (called *C*) of each protein to probe the similarity between the target and the template. The characteristic *C* is a string of vectors attached to amino acid sites. The site vectors may include the identity of the amino acid at the site, secondary structure, substitution probability in that site derived from multiple sequence alignment (profile), etc. The target characteristics include:

- **i.** A site-specific amino acid frequency (profile) generated from multiple sequence alignment of the target with homologous sequences from a standard sequence databases (NR -- http://helixweb.nih.gov/helixdb.php). The profile was created by PSI-BLAST with a single iteration and E value of 0.001.
- **ii.** The SABLE prediction of the secondary structure of each structural site from the amino acid sequence of the target [10,11].
- **iii.** The SABLE prediction of solvent-exposed surface-area from the amino acid sequence [10,11].

The three-dimensional co-ordinates of the template allow us to generate more characteristics:

- **i.** The raw sequence and profile of the template generated from the same database we used for the profile of the target.
- **ii.** The actual secondary structure of the template (extracted with the DSSP program [12]).
- **iii.** The actual exposed surface area of the template protein (extracted with the DSSP program [12]).
- **iv.** THOM2 contacts between structural sites (used in threading calculations) [13].

There are many ways of combining and comparing these characteristics between the target and the template to obtain matching scores that we call features (*F*) of pairs. For example, we may match a sequence with a sequence, secondary structure with a profile, test sequence fitness to contact maps (threading [14]), match predicted secondary structure of the target with actual secondary structure of the template, etc. The features are generally denoted by a four letter code such as SEQG (global sequence alignment), SEQL (local sequence alignment), TRDG (global threading), TRDL (local threading), TBLS (PSI-BLAST), TSCG (global secondary structure alignment), PSMG (sequence to profile matching - global), SRFL (local exposed surface area alignment) etc. Refer to the appendix of reference 2 for more details. To avoid repetition the features are not discussed in details in the present manuscript. We used 20 features in Phase 1, but have dropped 2 of these in Phase 2 (KMER and Contact Factor) because they have been insignificant, thus leaving 18 features.

Given two characteristics C_1 and C_2, there is a need for an alignment between the target and the template to generate a score, a scalar feature F_{12}. In LOOPP, we consider approximate alignments (BLAST), and exact (local or global) alignments determined by dynamic

programming. We also use constant or structurally dependent gap penalties [4]. We use four different types of scores when we compare the characteristics of the target and template to obtain a scalar feature value:

i. Raw scores. For input characteristics C_1 and C_2 the raw score is $S(\vec{C}_1, \vec{C}_2)$ where $\vec{C}$ denotes an extended character vector with the addition of deletions and insertions as required for an alignment. By convention the lower the energy (raw score) the better is the match. Raw scores are denoted by a _e extension to the four letter feature code, like TRDG_e, or SEQL_e.

ii. "Reverse score" [15] is computed as $S(\vec{C}_1, \vec{C}_{2r}) - S(\vec{C}_1, \vec{C}_2)$ where C_{2r} is the reverse characteristic input of the second protein. For example, if C_2 is the protein amino acid sequence $C_2 \equiv a_1 a_2 \ldots a_n$ then $C_{2r} \equiv a_n a_{n-1} \ldots a_1$. The reverse sequence provides an inexpensive measure of the deviation of the raw score from a match by chance. Higher values of the reverse score suggest a better match. Reverse scores are denoted by a _r extension such as TRDG_r, SEQL_r.

iii. The E-values from variants of BLAST and PSI-BLAST [16].

iv. The Z score was introduced already in the introduction and a more complete description is given below. These scores are denoted by a _z extension such as TRDG_z, SEQL_z.

$$Z(\overline{C}_1, \overline{C}_2) = \frac{S(\overline{C}_1, \overline{C}_2) - \left\langle S(\overline{C}_1, P\overline{C}_2) \right\rangle_P}{\sqrt{\left\langle S^2(\overline{C}_1, P\overline{C}_2) \right\rangle_P - \left\langle S(\overline{C}_1, P\overline{C}_2) \right\rangle_P^2}}$$

where the average $\langle S(\vec{C}_1, P\vec{C}_2) \rangle_P$ is over the optimal scores of the permutations of the characteristic C_2 at the amino acid sites. A single permutation is denoted by PC_2. The computational cost of the Z score was discussed in the introduction and it is between 100–1000 more expensive than the calculation of the raw score.

Z scores have been obtained for 12 of the 18 features in Phase 2. The raw energies, reverse energies and *Z* scores similarity measures contribute a total of 48 features. In addition, we also use two new secondary structure based features SSPOS and SSCOMP. SSPOS compares the position of the secondary structure elements between the template (actual) and target (predicted) whereas SSCOMP compares the amino acid composition of the actual and predicted secondary structures of the template and the target respectively. The prediction of the secondary structure of the target sites was made with the program SABLE [10,11]. Further, in Phase 2, we also generate atomic models for the chosen pairs and use the following potentials to derive the corresponding energies for these models. These energies are then used along with the features described earlier, in our training.

1. ENEALL, an all-atom potential [17]. This energy is a simple all-atom energy function (the distance dependence of contact is capture by a few steps). It was designed by Mathematical Programming optimization of a set of natives, approximate structures, and decoys.

2. TE13, [18]. This is one of the first energy functions computed with Mathematical Programming. It is a residue based contact potential with 13 steps to describe the distance dependence of the interaction of a pair of amino acids.

3. FREADY [19]. A new coarse-grained potential for proteins that includes two point masses per amino acid. The pair interactions have a complex distance and angular

dependence. It was derived by fitting distribution functions generated by Molecular Dynamics with the FREADY potential against distributions of the same variables extracted from the Protein Data Bank.

4. SIFT, a novel model assessment score, that combines multiple measures, including sequence independent mean radial distribution-based measure of packing [20] with sequence-derived solvent exposed surface area predictions [13,14] to assess the quality of a structural model [21]

Modeller [3] generates the atomic models based on templates chosen from phase 1 and alignments provided by the SSALN algorithm [4]. To assess the quality of Modeller output we compare three structures: 1) The native structure of the target sequence (which is known in the training), 2) The structure of the template (the structure of the homologous protein on which the modeling is based), and 3) The model of the target that Modeller built based on the template. Ideally the similarity (based on the TM score [7]) of structure 3 and structure 1 should be higher than the similarity of structure 2 and 1 since a refinement was performed. This is however not always the case. Sometimes structure 2 is closer to 1 than structure 3 to 1. This is especially unfortunate when the template and target structures are very close to begin with and Modeller produces structures that are farther from the target than the template.

The quality of the model is expected to be a monotonically decreasing function of the TM score between the template and the final structure. We therefore examine the TM alignment score between the template and the model. By examining the drift between the template and the structure generated by Modeller we have another independent assessment of the quality of the results. Thus, we have a total of 55 features that we use in our prediction model.

III. 2 Combining features to a single score

Given a set of features that measures the similarity between pairs of proteins (we have 55 features, some of them strongly correlated), we seek a computational model that uses these features to classify the pairs according to T (true matches) and D (decoy (false) matches). We have already seen in Phase 1 [2] that the tools of Mathematical Programming (MP) [22] are effective in this regard and hence we use MP in Phase 2 as well. The construction of the mathematical programming model is explained in detail in Phase 1 [2] and since we follow a similar approach, we avoid the repetition of the details and just provide the important points. We seek a score, $Q(i, j)$, to rank the matching quality of a pair of a target i and a template j, which is a linear combination of the features.

$$Q(i,j)=\sum_{\alpha=1,\ldots,K}\gamma_\alpha F_\alpha^{ij} \tag{1}$$

The γ_α are the unknown coefficients to be determined from the learning set with MP. The score function depends parametrically on the coefficients. In Phase 1 [2] we used scores with moderately more complex dependence on the features. For example a score was defined as a quadratic expansion of the features (e.g. the similarity score of the pair of proteins i, j (template and target) was $Q(i,j)=\sum_\alpha a_\alpha F_\alpha^{ij}+\sum_{\alpha,\beta} a_{\alpha,\beta}F_\alpha^{ij}F_\beta^{ij}$). In the present study we found that the more complex expansion was not enhancing the recognition capacity of the algorithms. The number of true hits was not increasing. We therefore remain with the simple form of equation (1)).

The training process requires sets of target and template pairs that are pre-classified as D (target-decoy) and T (target-true hit) pairs. It determines the linear coefficients γ subject to maximal margin and feasibility conditions [23,24]. We require that a match of a protein i with any T

template (say *j*) will have a better score than a match with a D template (say *k*). The requirement is written as an inequality $Q_T(i, j) - Q_D(i, k) > 0$ for a single comparison of the score $Q_D(i, k)$ of a decoy (*D*) with the score $Q_T(i, j)$ of a true match (*T*) to a target protein *i*. This inequality is linear and the set of all inequalities that we consider can be written compactly as

$$\begin{aligned} &Q_T(i,j) - Q_D(i,k) > 0 \quad \forall ij, ik \\ &\sum_{\alpha=1}^{K} \gamma_\alpha F_\alpha^{ij,T} - \sum_{\alpha=1}^{K} \gamma_\alpha F_\alpha^{ik,D} = \sum_{\alpha=1}^{K} \gamma_\alpha (F_\alpha^{ij,T} - F_\alpha^{ik,D}) > 0 \end{aligned} \tag{2}$$

where the indices *ij* and *ik* denote a true and a decoy pair respectively. The sum is over the elements of the scalar product, i.e. of the linear coefficients γ (to be determined) and the difference of the similarity measures of the two pairs. Numerically it is difficult to differentiate between a solution, which is slightly larger than zero and a solution with $\gamma = 0$ which is exactly zero. To set a scale for the values of the parameters, and to avoid a trivial solution, it is convenient to write

$$\sum_{\alpha=1}^{K} \gamma_\alpha (F_\alpha^{ij,T} - F_\alpha^{ik,D}) > 1 \quad \forall ij, ik \tag{3}$$

Equation (3) allows learning from negative (D) examples in addition to positive (T) examples. It therefore suggests a richer and more complete description of the data compared to classification algorithms that learn from positive examples only.

After significant trials of different functional forms of the similarity measures we were not able to find a single *Q* function that makes the problem feasible for all pairs of T and D and generalizes well to the test cases. It means that the set of features we have at present is insufficient to generate such a desired *Q* score. We believe that such a single function exists since a free energy surface that selects native folds was illustrated for many proteins. It is just that we do not know the proper functional form. Nevertheless, it is still possible to find a simple similarity measure that minimizes an error function and generalizes well to the test cases. The solution, however, does not solve all inequalities of Eq. (3). An alternative formulation is therefore required which is a slight adjustment of Eq. (3).

$$\begin{aligned} &\sum_{\alpha=1}^{K} \gamma_\alpha (P_\alpha^{ij,T} - P_\alpha^{ik,D}) > 1 - \eta_{ij,ik} \quad \forall ij, ik \\ &\eta_{ij,ik} \geq 0 \end{aligned} \tag{4}$$

$$\text{Subject to } \min\left(\sum_{ij,ik} \eta_{ij,ik}\right)$$

where the $\eta_{ij,ik}$ are (positive) slack variables that make the solution of the new set of inequalities feasible.

Apart from the inequalities generated as $Q_T^{ij} - Q_D^{ik} > 0$, as mentioned above, we also generate inequalities of the type $Q_T^{i1} - Q_T^{ij} > 0$, where the pair *i1* is the best true hit and the pairs *ij* are the remaining true hits (ranked 2 and below) for a given protein *i*. Eq. (4) then becomes:

$$\begin{aligned} &\sum_{\alpha=1}^{K} \gamma_\alpha (F_\alpha^{ij,T} - F_\alpha^{ik,D}) > 1 - \eta_{ij,ik} \quad \forall ij, ik \\ &\text{and} \\ &\sum_{\alpha=1}^{K} \gamma_\alpha (F_\alpha^{i1,T} - F_\alpha^{ij,D}) > 1 - \eta_{i1,ij} \quad \forall i1, ij \\ &\eta_{ij,ik} \geq 0 \end{aligned} \tag{5}$$

These inequalities are added to the inequalities obtained from $Q_T^{ij} - Q_D^{ik}$ pairs during learning. This is done to enforce learning in such a way that the best T is ranked first along with learning to rank the Ts above the Ds. In Phase 2, we are looking to identify the best hit in the top 1 or top 5 and we expect the $Q_T^{i1} - Q_T^{ij} > 0$ inequalities to help in this regard.

We use the Mathematical Programming (MP) solver, PF3 [22], which is tuned specifically to solve problems like Eqs. (4) and (5) frequently encountered in the field of bioinformatics and is based on the interior point algorithm. The actual numbers of pair comparisons (the number of inequalities) that we attempt to satisfy in Phase 2 is much less than Phase 1 since we have lesser number of T and D pairs. Hence, here we attempt to solve all the 10,922,967 (total from $Q_T^{ij} - Q_D^{ik}$ and $Q_T^{i1} - Q_T^{ij}$ pairs) of inequalities at one go. PF3 provides a solution to this problem within minutes using 20 computer nodes. The solution is then checked on all the inequalities of the set. The results, as stated above, are not exact. Not all inequalities are satisfied with the computed vector of coefficients γ. Another way to quantify our prediction system is by statistical significance of the predictions. Such a measure can be obtained from statistical theory (like in BLAST [25]) which we do not have for our multiple signals, or from numerical studies of a large number of examples. We can estimate the frequencies of T and D values quite accurately since the number of empirical observation is large. A score is considered significant if the probability that it is a D (Pr(*D*)) is less than or equal to 0.001. Since our learning set has been significantly cleaned up in Phase 1, we have a limited set in phase 2. Hence, using such a probability measure identifies approximately one D per protein in the dataset in Phase 2. We define the significance score as $SC \equiv 1 - \Pr(D)$ and this score is used in discriminating the Ts from the Ds in each step of our prediction model. In Phase 1 we used a tree of multiple linear scoring branches. We use a similar approach in learning Phase 2 as described below.

III. 3 A tree of *Q* scores for template identification

We have observed that a single *Q* score is insufficient in discriminating the Ts and the Ds. This is because some features like sequence similarity measures carry very strong signals that mask the signal from other features. We believe that if we remove sequences detected with one *Q* score, it is possible to learn another score that focuses on weaker (but nevertheless significant) signals. The idea is to use multiple scores where different *Q* are learned in sequence on a shrinking template database.

III. 3.1 The zero branch—We have learnt that PSI-BLAST is the most dominant signal and contributes significantly to the first branch. It is therefore convenient to use PSI-BLAST as a single feature in the first branch and identify all the strong PSI-BLAST pairs. This will help to filter out the strong PSI-BLAST pairs and thereby aid in recognizing the signal from the other features in the following branches. We had a similar zeroth branch in Phase 1 as well [2].

We generate a PSI-BLAST profile for the target sequence using the NR database http://helixweb.nih.gov/hilexdb.php (three cycles with an E-value threshold of 0.01). This profile is compared to the sequences of the templates to yield the desired score (The log of the E-value, called TBLS_e in the feature table of the Appendix). Using the significance measure discussed earlier, all pairs with significance score, *SC*, larger than 0.999 were declared hits. These are kept for the final analysis in which the selected pairs from all the branches are sorted and ranked to identify the best models. All pairs that are not hits are forwarded to the next branch.

An assessment of an optimal single score can be made using PSI-BLAST, which is widely employed in template detection. In Phase 2 we are using PSI-BLAST with reasonably high confidence level (E-value smaller or equal 0.001) and with that accuracy it detects less than half of the true hits we pick with the tree. Reducing the confidence level to E-value of 0.1 we find just in branch 2 a larger number of true hits (72,625 instead of 22,433). Unfortunately, the number of false hits increases significantly from 9,817 to 207,827 making the selection of top templates very difficult.

III. 3.2 Branch 1—Another feature of significance is the final score obtained in the previous LOOPP version that participated in CASP7 (LP7). This score is a single linear combination of a subset of features discussed earlier and in the appendix of ref. [2]. We find that this score has an important signal and can be used to find many of the strong hits beyond PSI-BLAST. We apply the same significance measure on all pairs forwarded from the PSI-BLAST branch and identify those with *SC* larger than 0.999 as hits and are chosen for final ranking. The rest, as usual, are forwarded to the next branch.

III. 3.3 Branches 2 through 6—With the pairs recognizable by straightforward PSI-BLAST and LP7 scores removed in the zero and first branches respectively, we seek in the second branch a prediction model linear with the features F_α described in the Appendix of reference [2]. The coefficients $\gamma_\alpha^{(2)}$ (the superscript (2) stands for branch 2) are determined from equation (5). We load all the 10,922,967 inequalities to solve for $\gamma_\alpha^{(2)}$. We then evaluate the scores for all the pairs and identify those with *SC* greater than 0.999. These are retained for final ranking and the rest are forwarded to the next branch. This procedure is repeated through branches 2 to 6 with a diminishing number of T and D pairs to learn from in each branch. We select predicted T pairs from each of these branches following the same *SC* procedure. All hits with scores above the threshold are forwarded for final evaluation. The pairs below threshold were forwarded to the next branch. As before, the threshold was set at a score value for which the probability of being a D was equal or smaller to 0.001. In Figure 2, we show the probabilities densities of T and D for the second branch. While there is a significant overlap for the two, there remains a significant tail of T pairs that can be picked with sufficient confidence. As we move from one branch to the next, the recognition capacity diminishes except in branch 6, where there is an increase in recognition capacity compared to the previous branches. This is due to the introduction of new features (FREADY, SIFT) in this branch. We stop our learning at the 6^{th} branch because, beyond this, the recognition capacity becomes insignificant i.e., the number of T pairs recognized with *SC* greater than 0.999 is negligible.

Like in Phase 1, we then tried using similarity measures derived from the features that are transformed into a uniform distribution which can be thought of as an alternative kernel [24] and also the quadratic expansion of the uniformly distributed variables. Although these were very useful in Phase 1, they were ineffective in Phase 2.

III. 3.4 Final ranking—At this stage we have a collection of predicted T pairs from the zero to six branches. Typical number of T predictions per protein varies from 1 to 200. Now, we need to rank the predicted T pairs that are pooled together from the different branches. We use

another linear score for the final sorting of the matches and we learn this the same way as we did in the last five branches – using the features mentioned earlier and solving for $\gamma_\alpha^{(2)}$ from the inequalities of the type given in Eq. 5, using PF3. The score thus obtained is a linear combination of the same features but the learning is done from the T and D pairs chosen from zero to six branches. Hence, the learning set here is minimal and is enriched with more Ts and less Ds. This also yields lesser number of inequalities to learn from. The linear score thus learnt is used for the final sorting of the pairs chosen from the zero to six branches of the tree. By far the largest coefficient of the linear combination of scores of branch 6 is that of LP7.

It is possible that re-learning the final ranking score using the pairs filtered from the zero to six branches and the same features lead to over-learning. Over-learning can lead to a prediction model that performs well on the training set but does not perform well on other independent test cases. We look into the performance of the prediction model on the test sets to provide insights into this potential problem.

IV. Results and Discussion

IV. 1 Description of the tree

The computed coefficients for all five branches of the tree plus the final ranking are summarized in Table 1a. For the purpose of comparison it is convenient to normalize the vector of coefficients such that $\sum_\alpha \bar{\gamma}_\alpha^2 = 1$ for all branches. We notice that some of the features are not used in all the branches (FREADY, LP7 score etc). These are mainly features that were derived much later and were not available during the initial stages of learning. In Table 1b we provide the product of the coefficient times the variance of the feature under consideration. This measure is another indicator of the potential contribution of a particular feature to the total score of a branch. If the variance of a feature is high, then it can (potentially) make a significant contribution even if the coefficient is small. Table 1b suggests the dominant contributions of only a few features (All-atom energy, SEQG_e, TRDG_e, TRSG_e, PSMG_e, PSML_e). However, this picture is somewhat misleading. It is possible to have a large variance and still only low recognition capacity. We know that the use of the other features (such as the Z-scores) is necessary in order to obtain a good recognition.

There are two factors that make the branches different from each other. First, the prediction spaces of the branches are different due to sequential elimination of T pairs. Second, the coefficient vectors are not the same. These differences are further enhanced in the final ranking branch since it uses the pairs selected from the 0–6 branches pooled together for learning. It is of interest to examine how similar the coefficients of the vectors are in the different branches and so we evaluate the scalar product of the normalized vectors between the different branches. This is presented in Table 2. We find that the coefficients of branches 2–5 are similar since their scalar products are in the range of 0.87–0.99 whereas those of branches 6 and final ranking are different from the rest (given by scalar products in the range 0.03–0.4). However, the scalar product of the coefficient vectors between branch 6 and final ranking is 0.8838 showing that they are very similar. Both scores are dominated by LP7.

Looking at the features providing the dominant signals in each branch, we find that in branches 2–5, the secondary structure based features, SSPOS and SSCOMP are very dominant. Apart from these, OPTM_e, SEQG_z, TRDG_z, TRSG_z, and TSSL_z provide signals in branches 2–5 whereas SEQL_z is strong in branches 2, 4 and 5. OPTM_e is a mixture of threading, secondary structure, and sequence alignment substitution tables; SEQG_z and SEQL_z are global and local sequence alignment Z scores respectively; TRDG_z and TRSG_z are threading-based global Z scores; TSSL_z is a combination of sequence, secondary structure and threading signals. In addition, profile-sequence matching score, PSML_z is dominant in

branches 2 and 3 whereas PSMG_z is dominant in branches 4 and 5. Further, threading scores TRDL_z and TSSG_z are dominant in branch 3 and branches 4 & 5 respectively.

Branch 6 is different because new features are introduced which provide dominant signals. LP7 score, FREADY [19] and SIFT [20,21], all are new features that are dominant in this branch. In the branch of final ranking we find that the weight of the LP7 score is very high thus skewing the other weights. FREADY is a coarse grained energy computed from the final atomically detailed model. SIFT is a model assessment score that combines sequence dependent secondary structure and expose surface area prediction with mean radial distribution function-based assessment of packing, which is independent of the sequence of the template. LP7 is a linear combination of the following features: Protein length, SEQL_e, TRDL_r, TRSG_e, TRSG_r, PSML_r, SRFG_e, OPTM_e, TSSG_r, TRSL_e and TSCL_e.

Although the dominant LP7 score does not contain any *Z* scores, note the predominance of *Z* scores rather than raw or reverse scores in the other features that have *Z* scores evaluated. This shows the significance of *Z* scores and validates the computational time spent in evaluating these expensive features. Also of significance is the absence of PSI-BLAST related features (TBLS, TBSS and SBLS) that were the most dominant signals in the branches of Phase 1 tree. Although we use PSI-BLAST (TBLS_e) in branch zero and we recognize maximum number of T pairs in this branch, we find that in the rest of the branches and in the final ranking, PSI-BLAST plays a minimal role. Profile matching scores PSMG_z and PSML_z along with simple global and local alignment scores SEQG_z and SEQL_z are the only sequence-based features making any kind of contributions to these branches. In some manner, this proves that using such a tree-method eliminates the dominant PSI-BLAST signals in branch 0 and helps in picking up the signals from the other features in the later branches. This also validates our tree-based algorithm and thus enables us to identify the hits that cannot be recognized with PSI-BLAST alone. However, the drawback is that valuable PSI-BLAST hits can be lost in the final ranking due to other features being dominant there. We discuss this in detail in a later section.

IV. 2 Application of the model

The developed prediction tree is applied as follows. The target sequence is first examined with branch 0. PSI-BLAST scores are computed for any target-template pairs where the templates are taken from the complete Protein Data Bank (PDB). LOOPP has a standard database that is used by all other branches, however PSI-BLAST is so efficient to compute that we probe the complete PDB with it. If the probability of observing a false hit is smaller than 10^{-3} accept the match as a hit and store the hit in the list of candidates. If the structure is found in the standard LOOPP database, we remove it from the set of the structures that we need to examine in the next branch. In the next branch we seek hits of the target with templates that were not detected before. Hence the data set we examine is a subset of the total. The next branches do not consider the hits of previous branches. At the end of the process (when the last branch delivers its hits to the pool of candidates) all the hits are ranked against each other. The final ranking is done with another Mathematical Programming score.

IV. 3 Evaluating performance

Figure 3 shows the comparative statistics of the number of hits identified per protein in phase 2 with respect to phase 1. The plot shows a histogram of the number of proteins as a function of number of hits forwarded from Phase 1 and the corresponding number of hits detected in Phase 2. We see that most of the proteins with 0 and 10 hits that were identified in Phase 1 are also identified in Phase 2. There are 899 proteins where no hits are detected in phase 2 irrespective of the number of hits forwarded from phase 1. Of these, there are a handful of cases (77), where proteins with more than 10 hits forwarded from phase 1 have zero hits

detected in phase 2. The remaining proteins have at least one or more hits identified in phase 2 and a few have as high as hundred hits identified in both phase 1 and phase 2.

As in Phase 1, we evaluate the number of T pairs identified and the number of proteins with at least one hit and (or) T hit identified in Phase 2. These results are tabulated in Table 3. A few points to note:

- **a.** The total number of T pairs solved by the tree in the LS is 162,762 out of 209,090 forwarded from Phase 1, which is 78%. Since 91% of the proteins have at least one T hit (see below) this performance is actually better than it looks from first sight.
- **b.** PSI-BLAST provides the most dominant signal, detecting 62,419 (30%) of T pairs, followed by the LP7 score with 52,727 (25%) T pairs. The remaining branches put together identify 54,373 (26%) percent of T pairs.
- **c.** The tree model that we generated on the training set generalizes quite well on the test sets with comparable performance on TS1, with 78.6% T pairs identified over all (30,943 detected out of 39,364 forwarded from Phase 1). The performance of LP7 score is significantly weaker in TS1 compared to LS. This poorer performance of the LP7 branch in TS1 is compensated by slight increases of the performances of other branches.
- **d.** The performance in TS2 is comparatively lower than LS and TS1 with 768 T pairs detected out of 1125 forwarded from Phase 1 (68.3%).
- **e.** Table 3 also provides the number of proteins with at least one hit identified (may be T or D) and the number of proteins with at least one T hit identified. The tree model identifies at least one hit in 91% of proteins (11,420 out of 12,527 forwarded from Phase 1) and detects at least one T hit in 92% of proteins (10,795 out of 11,694 forwarded from Phase 1) in the LS. TS1 performs comparably with 94% of proteins with at least one hit and 95% of proteins with at least one T hit identified. However, TS2 performs a bit lower with 85% and 81% respectively. Since TS2 is the smallest set, it is possible that statistical fluctuations cause the difference.

Therefore, the overall performance of the tree is similar in LS and TS1 although, it is slightly diminished in TS2. Further, the tree is able to identify T pairs in more than 90% of the proteins in both LS and TS1 thus enhancing the prediction capacity of LOOPP. The performance of the final ranking on the hits detected by the tree is elucidated as follows.

In Phase 1, we used the number of T pairs identified and proteins with at least one T pair detected to evaluate our performance of target recognition. These were useful measures in Phase 1 because we forwarded top 200 hits from Phase 1 to Phase 2, where we carried out more accurate and expensive calculations. In Phase 2 we are looking to identify the best models and hence we need to evaluate ranking. Ultimately, we need to identify the best model or the top 5 best models (as in CASP). Since we know the native structures in this case, we already know the best model based on the RMSD of the model generated by LOOPP with respect to the native structure. We then rank the models identified by Phase 2 tree based on the linear score from final ranking. Then we check if our identification/ranking scheme identifies the actual rmsd-wise best model in the top 1 or top 5 positions. Additionally, we also check whether the pairs identified in top 1 are T or D (at least one in top 5 is a T) as classified by our initial scheme. These results are tabulated in Table 4 for both LS and TS. For comparison, we provide the same results for LP7 as well.

Table 4 summarizes the important results of this paper discussed below:

1. In the LS, the tree identifies the best hit in the top 5 in 90% of the proteins and the best hit in the top 1 in 59% of proteins. Further, 94% of proteins in the LS have a true hit in the top 5 and 90% have a true hit in the top 1.
2. Compared to LP7, the performance of the tree is remarkably better since LP7 identifies the best hit in the top 5 in 70% of the proteins and in top 1 in 43 % of proteins of the LS. Similarly, LP7 identifies a true hit in top 5 in 89% of proteins and top 1 in 84% of proteins. Hence, the tree has significantly improved template recognition compared to LP7.
3. These numbers for TS1 are similar to that of the LS, thus eliminating the doubts about over-learning. However, in TS2, which is the CASP7 dataset, we see that the performance of the tree is lower compared to LS and TS1, consistent with the other results as well.

Figure 4 shows a histogram of the number of proteins versus the TM score of the top model and the best model in the top 5 as identified by Phase 2. There are more proteins where the top models have tm scores ≥ 0.65. However, there are 1454 proteins where the top models have tm scores < 0.65 and 1303 proteins where the best model in the top 5 have tm scores < 0.65. Table 5 shows the T and D classification of these models. There are more Ts than Ds in both tm ≥ 0.65 and tm < 0.65 categories. However, the Ts are way higher for tm ≥ 0.65 than in tm < 0.65 showing the overall enrichment of the true hits in Phase 2. The Ts in tm < 0.65 category and the Ds in tm ≥ 0.65 category are borderline cases, where there are discrepancies between Modeller and TMalign.

IV. 3 CASP 8 performance

LOOPP server participated in CASP 8 structure prediction experiment during the summer of 2008 (http://predictioncenter.gc.ucdavis.edu/casp8/index.cgi). The LOOPP server accepts sequence electronically, identifies the templates, generates atomically detailed models, scores them and e-mails back the results and top scoring models. It does not use results from any other servers while meta-servers were included in the ranking of CASP8 and in the discussions below. We have not done as well as we hoped and we are learning the results at present. Unfortunately, during CASP8 the LOOPP server was not stable. We updated its databases well into the exercise and found during the exercise several bugs. To ensure that the results reported are meaningful and reproducible we report them twice. One set for the actual LOOPP server and a second set for the stable version of LOOPP that was achieved towards the end of the competition. We analyze in more details the models of the stable version only.

Among the groups that submit at least 100 targets of the total of 115 targets (there were 66 such groups) LOOPP is ranked in the lower third. The average TM score over all targets of the model ranked first in LOOPP CASP submission was 0.612 compared to the best group by our assessment, the Zhang server, with TM score of 0.702. The rank of the first model is 51 compared to other servers. If the stable LOOPP is considered, the average TM score of the first model of LOOPP climbs to 0.647 and the rank to 45.

LOOPP is doing better when ranking the best model out of the five submissions to CASP8. The average TM score of actual LOOPP submission was 0.671 (rank 44, the best average was again the Zhang server with 0.719). When the stable version of LOOPP was considered the score was 0.691 and the rank was 26. While LOOPP requires considerable improvement perhaps the most striking observation for us is the high density of groups in the neighborhood of 0.6–0.7 TM scores suggesting that the differences between groups are not as large as one may suspect. For comparison the TM score of the group ranked 58 was 0.604.

To gain further insight to the performance of the algorithm we analyze in more details the first 63 targets. There are 13362 models generated for the 63 targets. Of these, LOOPP Phase 2 identifies 1560 as hits for 61 targets (2 proteins have no hits identified). A TM score between the native and the model, which is better or equal to 0.5, is considered a true hit. The models include 1951 true hits of which LOOPP identify 1393. When the TM scores of the template to the native structures are examined we get 2169 true hits.

Further, of the 1560 hits identified, the alignment and the model building (MB) makes 672 models worse than the template, of which 69 are bad templates to begin with. So, model building from template "spoils" 603 good templates out of the whole 1560 hits. Of the 603, 39 are unacceptable (less than 0.5 TM), whereas 564 are still acceptable (greater than 0.5 TM) although the template was better than the model. Similarly, MB makes 885 models better than the template, of which, 58 models are still bad in spite of improvement compared to the template. 800 models were already good templates to begin with. There are 27 cases, where MB takes a bad template and makes a model out of it. There are 3 cases where MB does not affect the template/model at all. Also, there are 124 hits, where MB makes the model worse by more than 0.05 tm score when compared to the template, of which 116 are from good templates.

Overall performance—We find the best model in top 5 hits 77% percent of the time, the best model in top 1: 43%, best template in top 5: 80%, and best template in top 1: 38% of the time. The overall performance is lower compared to what we observed for the training, test and CASP7 sets. It is possible that the additions to the Protein Data Bank that happened after 6/28/2005 are sufficiently different that re-training of the model is required (for CASP8 we updated the databases but not the prediction model).

A few concrete observations are discussed below

1. Targets 397 and 465 have no hit identified. 397, because there seems to be no good template (in Phase 2 input). 465 has a single reasonable template (TM approx. 0.5), which the Phase 2 tree fails to identify.
2. In thirteen cases (out of 61), we miss the best hit. 6 of them because there is no good template (in Phase 2 input), 2 of them because of unsuccessful model building and 5 because Phase 2 tree fails to identify the reasonable templates.
3. Our learning de-emphasizes target and template matches with high sequence similarity. As a result the model that we finally developed misses several trivial PSI-BLAST hits during CASP8. The PSI-BLAST hits were detected at the appropriate zero branch. However, when all the hits were collected, the hits from other branches mask the PSI-BLAST hits. We will need to rectify the model and probably assign a special protocol for sequence with high PSI-BLAST score.
4. T0413 is a difficult target, where we perform well with our model ranked as the second best server model by CASP evaluators. The target is a poly(3-hydroxybutyrate) depolymerase with an α/β hydrolase fold (PDB ID: 3D0K). The CASP evaluators have listed 20 close templates in the PDB for this target, most of them within 2.75 Å C-alpha RMSD from the target native. Most of these are esterases with α/β hydrolase fold, 1JJID, a carboxylesterase, being the top one with 2.4 Å CA RMSD from the target native. Although, we do not identify any of the top templates as identified by the CASP evaluators, we do identify a putative esterase with an α/β hydrolase fold as the top template (1PV1A), which has a 2.6 Å CA RMSD from 3D0K. Further, our top template, 1PV1A, has a better TM score with the target native (0.65) as compared to the top template provided by the CASP evaluators, 1JJID (0.58). Since our learning is based on TM scores rather than RMSDs, our server follows well our training and

picked 1PV1A ahead of 1JJID. We also find that the other top servers for this target (Zhang server, Robetta) do not pick any of the top templates listed by the CASP evaluators as well as 1PV1A, identified by LOOPP. The FEIG server, which performed comparably to LOOPP does not provide the template information for comparison. Further, the alignment of 1PV1A to the target has 222 out of 304 (73%) residues aligned within ± 4 residues from that of the TM alignment of 1PV1A with 3D0K. Further, the model to native TM score is 0.65, similar to the template to native TM score. Therefore, the success of the prediction for this target is mainly due to the success in template identification along with a reasonably good alignment.

To further elucidate some successes and failures we present the structural alignments of the best predicted LOOPP models from the top 5, with the native structures of four targets – T0428, T0415, T0411 and T0472 (figure 5). These four targets have been chosen to bring forward our hits and misses. T0428 is one of our best predictions, where we identify the best template available and produce a very good model with a TM score of 0.95 with the native. T0472 is one of the worst misses, since there is a very good PSI-BLAST hit for this target, which LOOPP fails to identify and hence the model is also very poor (0.27 tm score with the native). T0415 and T0411 lie in between these two extreme examples. In T0415, although we identify the best template, our model is not the best (0.73 tm score with the native) and in T0411, although we do not identify the best template, our model is reasonably good (0.71 tm score with the native). There are other cases, like T0419 and T0478, where the target itself is a tough one with no reasonable template in the database and hence the models are also poor.

V. Concluding remarks

We present a computational model for selecting templates from the protein data bank and building atomically detailed structures for target sequences. It is illustrated that both in target selection and in model building there is a significant “bleeding” and loss of good templates. The losses are due to mis-classification of good templates and to suboptimal refinement. Nevertheless, because of the growth of database sizes and coverage we have in many cases multiple hits for a single target. Even if a few good templates are missed, in many cases there are other good templates to fill in. If we measure the success of the algorithm by its ability to find a sound template, then the algorithm is quite successful as is evident in table 4. It therefore seems that future research should focus on the step of refinement.

Acknowledgments

This research was supported by NIH grant GM067823 to Ron Elber. Brinda Kizhakke Vallat was supported by a fellowship Human Frontier Science Program Long Term Fellowship: LT00469/2007-L.

References

1. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Research 2000;28(1):235–242. [PubMed: 10592235]
2. Vallat BK, Pillardy J, Elber R. A template-finding algorithm and a comprehensive benchmark for homology modeling of proteins. Proteins 2008 Aug 15;72(3):910–928. [PubMed: 18300226]
3. Eswar N, Mari-Renom M, Webb B, Madhusudhan MS, Eramian D, Shen M, Pieper U, Sali A. Comparative Protein Structure Modeling with Modeller. Current Protocols in Bioinformatics 2006;5.6:5.6.1–5.6.30. [PubMed: 18428766]
4. Qiu J, Elber R. SSALN: An alignment algorithm using structure-dependent substitution matrices and gap penalties learned from structurally aligned protein pairs. Proteins-Structure Function and Bioinformatics 2006;62(4):881–891.

5. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP - a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology 1995;247(4):536–540. [PubMed: 7723011]
6. Chun-Nam JY, Thorsten J, Elber R. Support Vector Training of Protein Alignment Models. Lecture notes in bioinformatics: RECOMB 2007. 2007
7. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Research 2005;33(7):2302–2309. [PubMed: 15849316]
8. Moult J. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Current Opinion in Structural Biology 2005;15(3):285–289. [PubMed: 15939584]
9. Kryshtafovych A, Venclovas C, Fidelis K, Moult J. Progress over the first decade of CASP experiments. Proteins-Structure Function and Bioinformatics 2005;61:225–236.
10. Adamczak R, Porollo A, Meller J. Accurate prediction of solvent accessibility using neural networks-based regression. Proteins-Structure Function and Bioinformatics 2004;56(4):753–767.
11. Adamczak R, Porollo A, Meller J. Combining prediction of secondary structure and solvent accessibility in proteins. Proteins-Structure Function and Bioinformatics 2005;59(3):467–475.
12. Kabsch W, Sander C. Dictionary of protein secondary structure – Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983;22(12):2577–2637. [PubMed: 6667333]
13. Meller J, Elber R. Linear programming optimization and a double statistical filter for protein threading protocols. Proteins-Structure Function and Genetics 2001;45(3):241–261.
14. Meller J, Elber R. Protein recognition by sequence-to-structure fitness: Bridging efficiency and capacity of threading models. Computational Methods for Protein Folding 2002;120:77–130.Advances in Chemical Physics
15. Karplus K, Karchin R, Shackelford G, Hughey R. Calibrating E-values for hidden Markov models using reverse-sequence null models. Bioinformatics 2005;21(22):4107–4115. [PubMed: 16123115]
16. Altschul SF, Madden TL, Schaffer AA, Zhang JH, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 1997;25(17):3389–3402. [PubMed: 9254694]
17. Jian, Qiu; Ron, Elber. Atomically detailed potentials to recognize native and approximate protein structures. Proteins, Structure, Function, and Bioinformatics 2005;61:44–55.
18. Dror, Tobi; Ron, Elber. Distance dependent, pair potential for protein folding: Results from linear optimization. Proteins, Structure Function and Genetics 2000;41:40–16.
19. Peter, Májek; Ron, Elber. A coarse grained potential for fold recognition and molecular dynamics simulations of proteins. to be submitted
20. Adamczak R, Meller J. On the Transferability of Folding and Threading Potentials and Sequence-Independent Filters for Protein Folding Simulations. Molecular Physics 2004;102(11–12):1291–1305.
21. Adamczak R, Meller J. Efficient and Accurate Protein Model Quality Assessment with Structural Profiles. to be published
22. Wagner M, Meller J, Elber R. Large-scale linear programming techniques for the design of protein folding potentials. Mathematical Programming 2004;101(2):301–318.
23. Meller J, Wagner M, Elber R. Maximum feasibility guideline in the design and analysis of protein folding potentials. Journal of Computational Chemistry 2002;23(1):111–118. [PubMed: 11913376]
24. Cristianini, N.; Shawe-Taylor, J. An introduction to support Vector Machines and other kernel based learning methods. Cambridge: Cambridge University Press; 2000.
25. Karlin S, Altschul SF. Applications and statistics for multiple high-scoring segments in molecular sequences. Proceedings of the National Academy of Sciences of the United States of America 1993;90(12):5873–5877. [PubMed: 8390686]

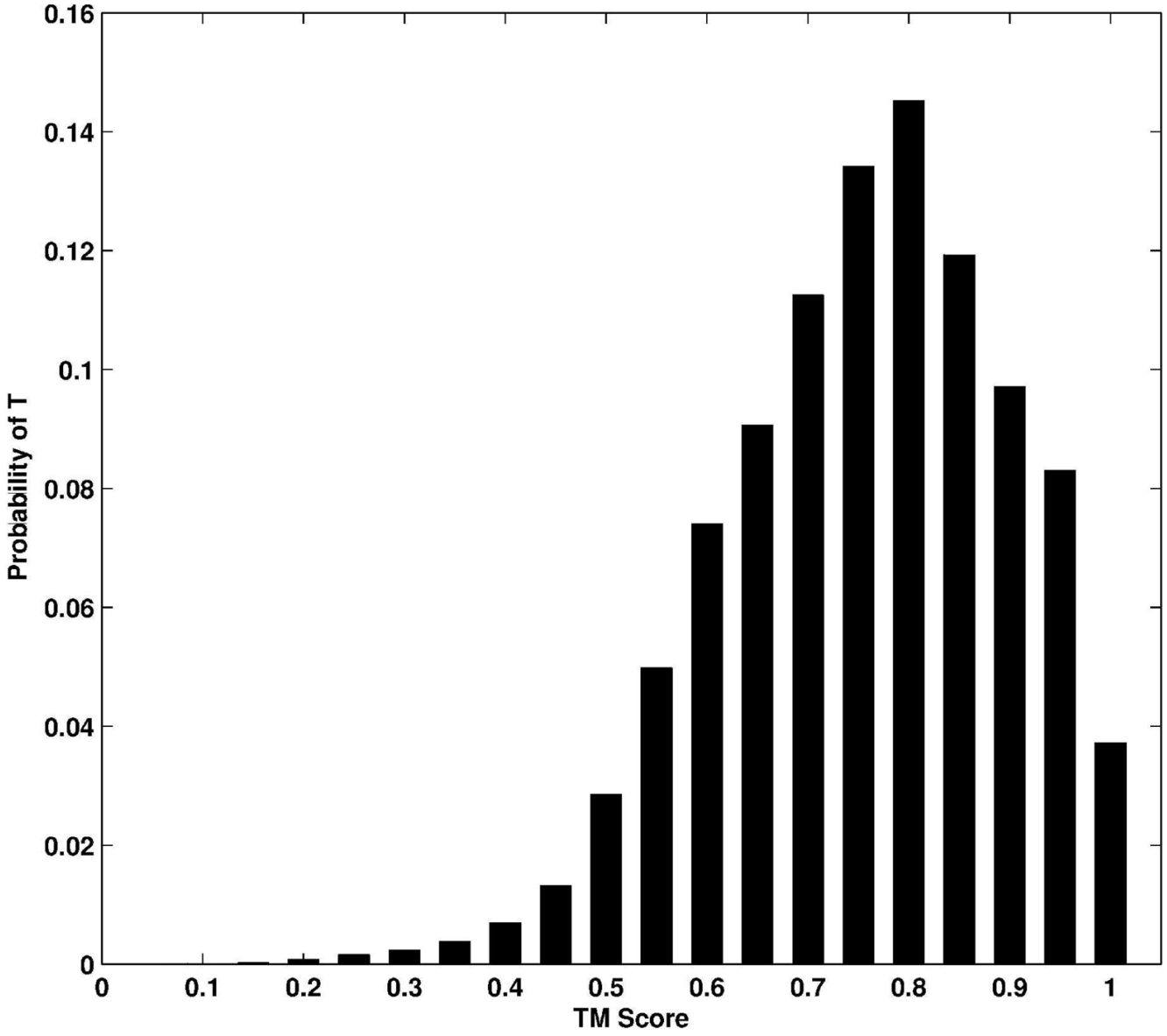

Figure 1.
Plot of the probability of finding a true hit as a function of TM score in Phase 2 in the learning set (LS).

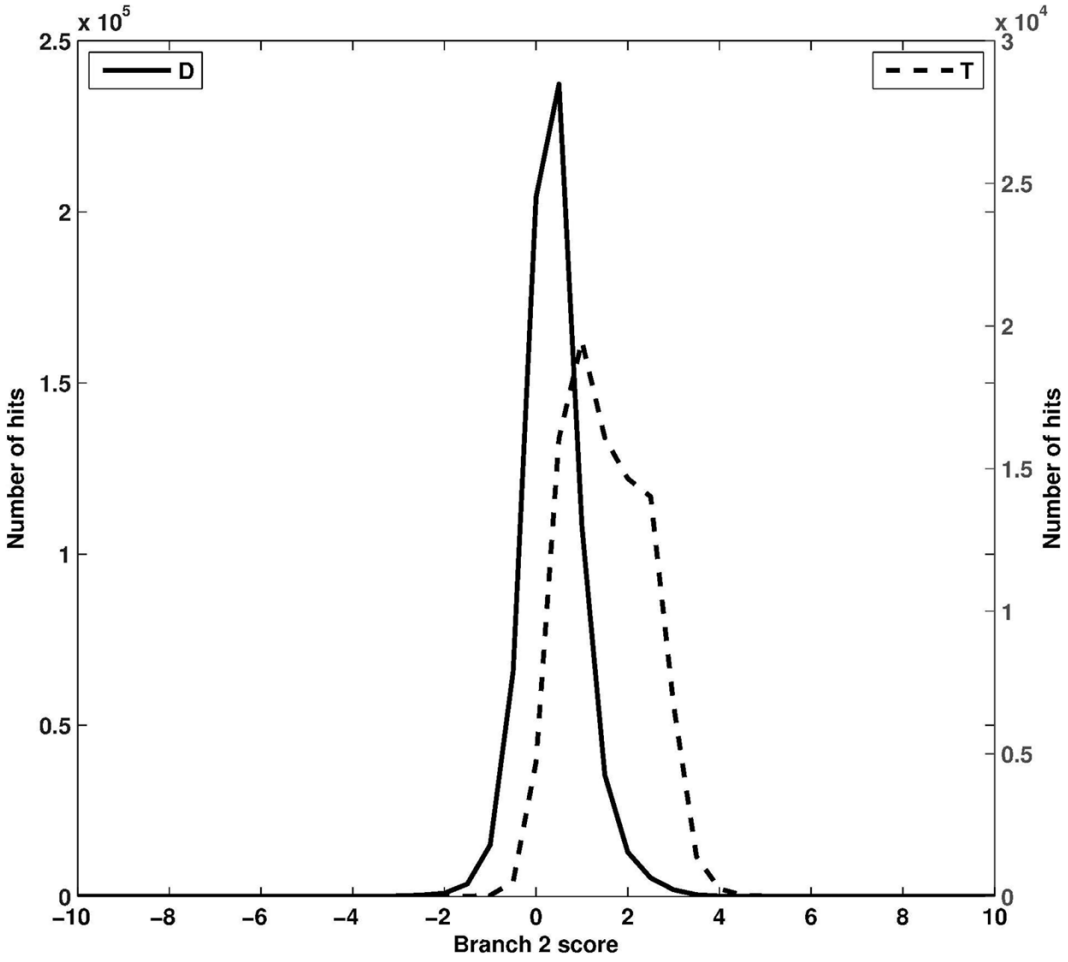

Figure 2.
Overlap of the probability density of the scores for true (T) and false/decoy (D) pairs of the second branch of the LS (first branch using mathematical programming).

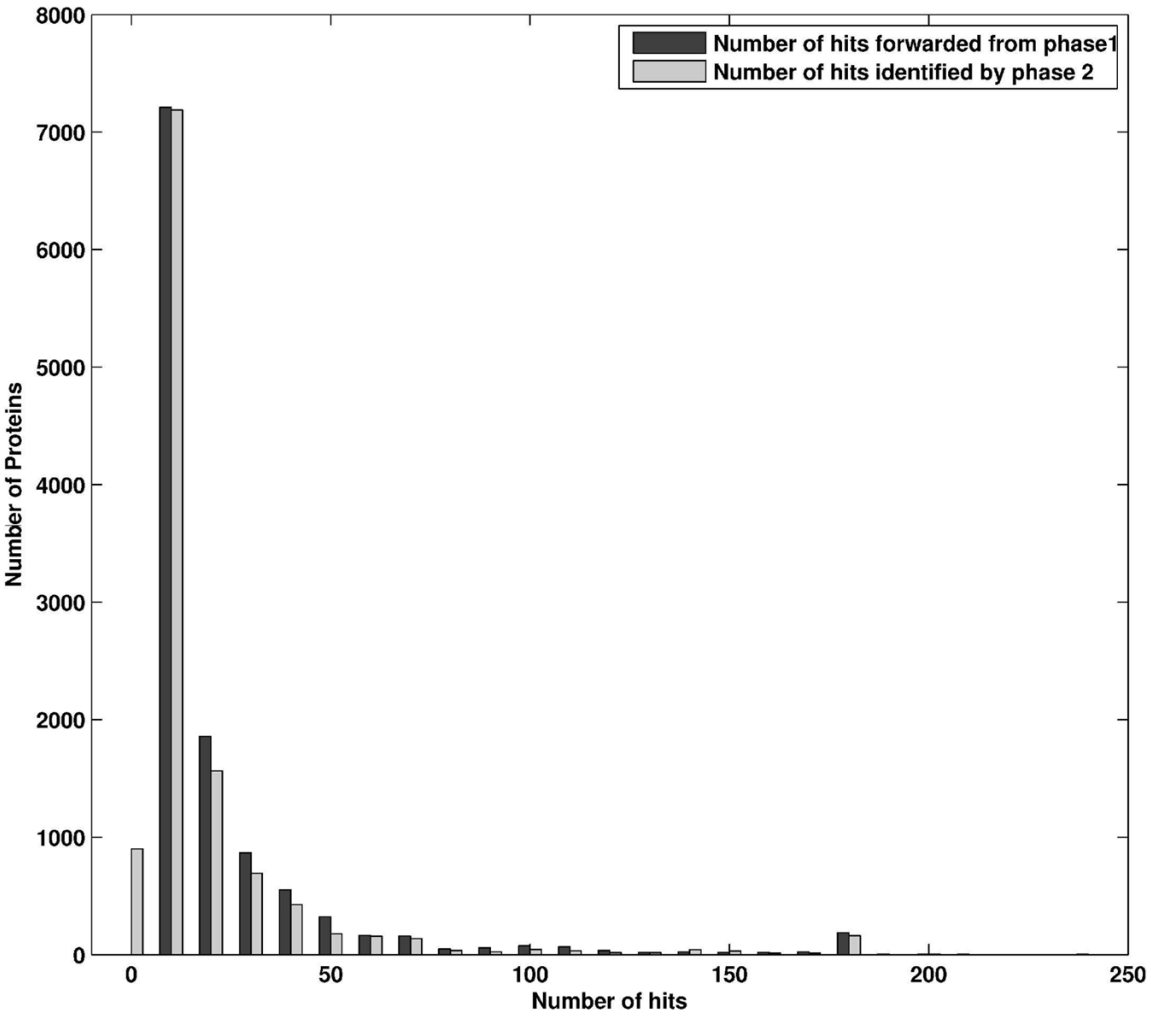

Figure 3.
Histogram of the number of proteins as a function of the number of pairs forwarded from Phase 1 and the number of pairs identified by Phase 2 tree.

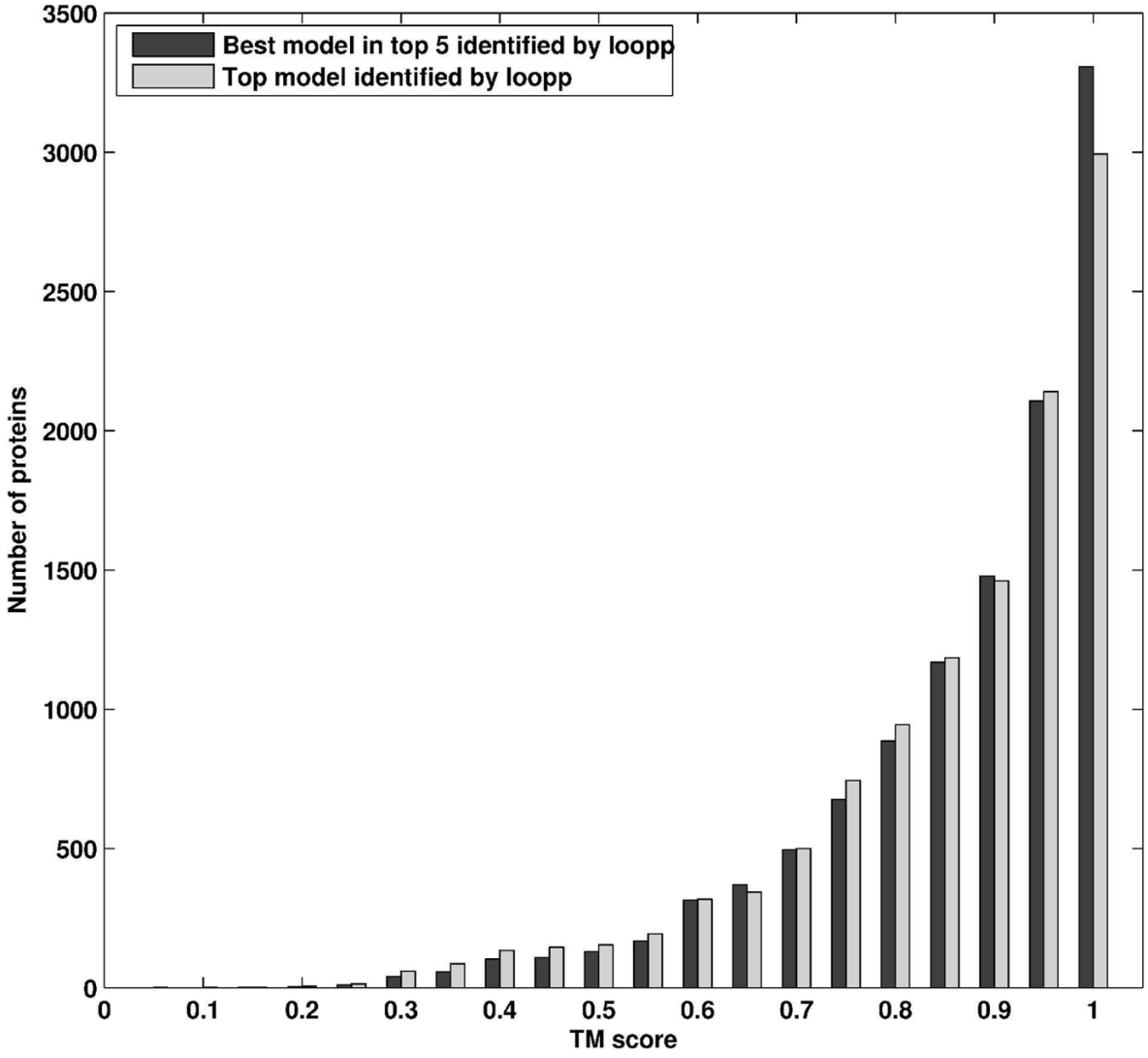

Figure 4.
Plot of the number of proteins as a function of TM scores of the top 1 hit and the best hit in the top 5 hits identified by LOOPP Phase 2 tree + final ranking scheme.

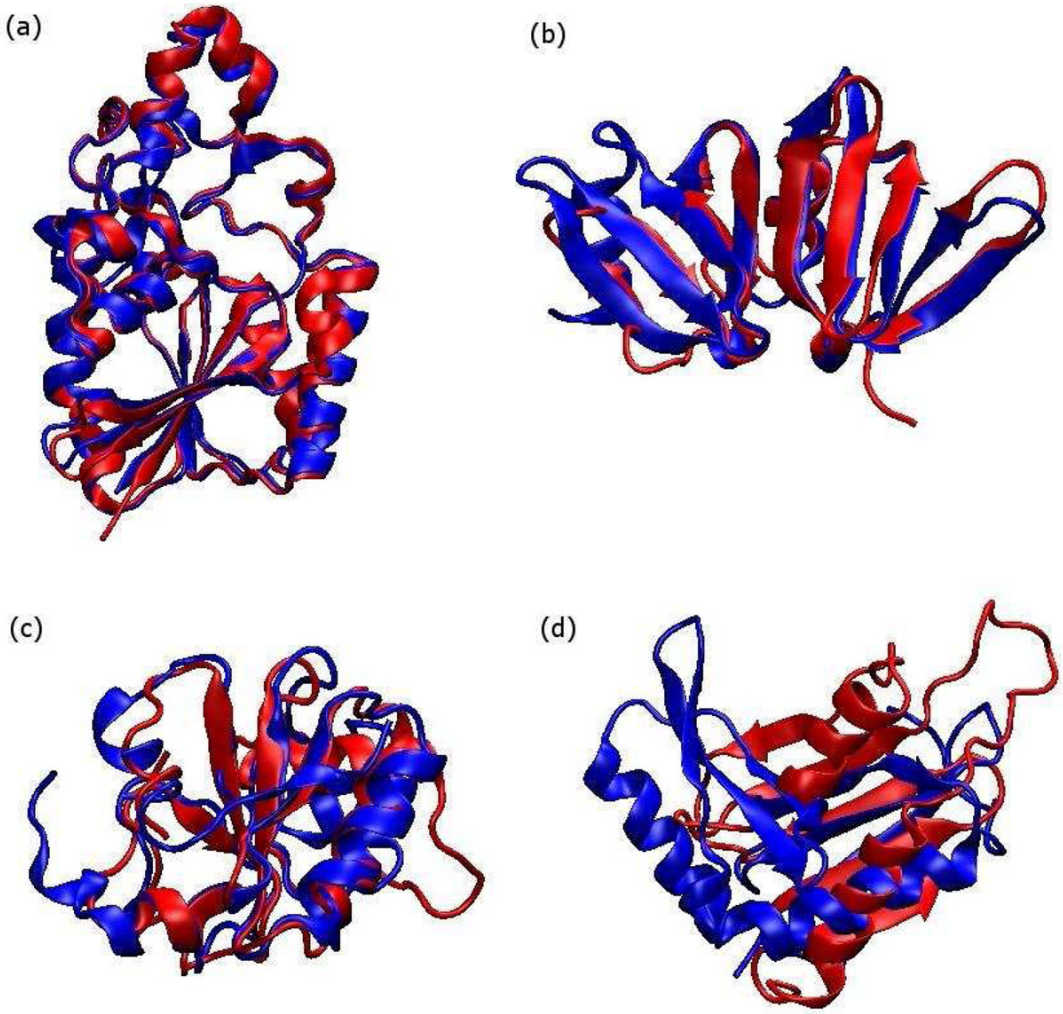

Figure 5.
Structural alignment of the best LOOPP model with the native structure for four CASP 8 targets: (a) T0428, where the best template is identified and we get a very good model with a tm score of 0.95 with the native, (b) T0415, where the best template is identified, however the model is only reasonably good with a tm score of 0.73 with the native, (c) T0411, where the best template is not identified in the top 5, however we obtain a reasonably good model with a tm score of 0.71 with the native, and (d) T0472, where the best template is not identified in the top 5 because we miss the good PSI-BLAST hit and the best model predicted by LOOPP is poor when compared to the native with a tm score of 0.27.

Table 1

Numerical weight coefficients for the different features in the different branches. Table 1a the coefficient that are used in the scoring. Table 1b the multiplication of the coefficients by the variance of values that a particular feature takes. The entries in 1b estimate the actual contributions of different features to the scores. The entries in 1a can be used to reproduce our prediction.

Table 1a: Normalized weights learnt for the features in different branches of the tree

Feature	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
LP7 Conf	--[#]	--	--	--	0.0082	0.0023
LP7 Score	--	--	--	--	0.8706	0.9962
Length	--	--	--	--	−0.0081	−0.0004
Seqident	--	--	--	--	0.0004	0.0005
All-atom energy[*]	−0.0062	−0.0074	−0.0057	−0.0054	0.0007	0.0012
TE13 energy[*]	−0.0005	−0.0005	−0.0024	−0.0025	−0.0002	0.0001
OPTM_r	0.0581	0.0422	0.0356	0.0346	0.0020	0.0004
OPTM_e	−0.2151	−0.2566	−0.3049	−0.3180	−0.0360	−0.0002
SEQG_z	0.1948	0.2527	0.2693	0.2597	0.0232	−0.0065
SEQG_r	−0.0014	−0.0013	−0.0006	−0.0003	−0.0001	0.0000
SEQG_e	0.0120	0.0140	0.0157	0.0161	0.0015	0.0000
SEQL_z	−0.1059	−0.0029	0.1778	0.1967	−0.0171	−0.0145
SEQL_r	0.0005	0.0018	0.0018	0.0019	0.0001	0.0000
SEQL_e	−0.0133	0.0034	0.0284	0.0302	0.0005	0.0001
TRDG_z	0.1836	0.1642	0.1706	0.1610	0.0098	−0.0002
TRDG_r	0.0307	0.0525	0.0226	0.0196	0.0009	0.0001
TRDG_e	0.0888	0.0887	0.0688	0.0659	0.0021	0.0000
TRDL_z	−0.0758	−0.1188	−0.0531	−0.0455	−0.0082	0.0005
TRDL_r	−0.0071	0.0224	0.0290	0.0330	0.0026	−0.0002
TRDL_e	−0.0101	0.0013	0.0392	0.0488	0.0014	0.0000
TRSG_z	−0.2374	−0.2999	−0.2886	−0.2733	−0.0230	−0.0130
TRSG_r	−0.0285	−0.0251	−0.0133	−0.0114	−0.0006	0.0000
TRSG_e	−0.0894	−0.0935	−0.0773	−0.0741	−0.0021	−0.0001
TRSL_z	0.0402	0.0378	−0.0094	−0.0237	−0.0142	−0.0167
TRSL_r	−0.0035	−0.0187	−0.0154	−0.0155	−0.0009	0.0000
TRSL_e	−0.0090	−0.0248	−0.0401	−0.0477	0.0002	0.0018
TSSG_z	0.0101	0.0293	0.1186	0.1204	−0.0022	−0.0081

Table 1a: Normalized weights learnt for the features in different branches of the tree

Feature	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
TSSG_r	−0.0098	−0.0121	−0.0175	−0.0177	−0.0014	0.0000
TSSG_e	−0.0117	−0.0003	0.0091	0.0084	−0.0009	0.0000
TSSL_z	0.2439	0.2490	0.1739	0.1651	−0.0057	−0.0224
TSSL_r	−0.0310	−0.0306	−0.0241	−0.0232	−0.0008	−0.0002
TSSL_e	0.0736	0.0739	0.0296	0.0267	0.0050	−0.0001
TSCG_r	−0.0016	0.0140	0.0130	0.0136	0.0012	0.0001
TSCG_e	0.0111	0.0069	0.0275	0.0294	0.0018	0.0015
TSCL_r	−0.0225	−0.0412	−0.0232	−0.0226	−0.0030	0.0001
TSCL_e	0.0160	0.0132	0.0214	0.0218	0.0077	0.0079
PSMG_z	−0.0182	−0.0708	−0.1618	−0.1668	−0.0265	−0.0311
PSMG_r	0.0008	0.0013	0.0009	0.0008	0.0001	0.0000
PSMG_e	−0.0068	−0.0103	−0.0151	−0.0161	−0.0002	0.0007
PSML_z	0.2211	0.1141	0.0441	0.0436	−0.0223	−0.0202
PSML_r	−0.0038	−0.0016	0.0018	0.0030	0.0001	0.0000
PSML_e	0.0160	−0.0055	−0.0160	−0.0169	−0.0005	0.0028
SRFG_z	0.0486	0.0652	0.0254	0.0147	−0.0132	−0.0265
SRFG_r	−0.0005	0.0061	0.0021	0.0016	0.0001	0.0000
SRFG_e	0.0054	0.0034	−0.0066	−0.0074	−0.0002	−0.0001
SRFL_z	0.0693	0.0929	−0.0066	−0.0158	−0.0396	−0.0344
SRFL_r	0.0133	0.0098	0.0220	0.0229	0.0023	−0.0001
SRFL_e	0.0033	0.0174	0.0109	0.0104	0.0023	0.0026
TBLS_r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
TBLS_e	−0.0041	−0.0034	−0.0030	−0.0027	−0.0003	0.0000
TBSS_r	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
TBSS_e	−0.0035	−0.0029	−0.0036	−0.0033	−0.0004	0.0000
SBLS_r	−0.0098	−0.0016	0.0042	0.0008	−0.0008	−0.0002
SBLS_e	0.0215	0.0143	0.0015	0.0027	0.0005	0.0000
SSPOS	−0.6932	−0.7310	−0.7109	−0.7313	−0.0125	−0.0040
SSCOMP	0.4246	0.2769	0.2937	0.2430	−0.0074	0.0023
FREADY*	--	--	--	--	0.3537	0.0278

Proteins. Author manuscript; available in PMC 2010 September 1.

Table 1a: Normalized weights learnt for the features in different branches of the tree

Feature	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
SIFTscore*	--	--	--	--	0.2563	0.0174
TM_mod_tem*	--	--	--	--	0.0618	0.0249

Table 1b

Feature	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
LP7 conf	0	0	0	0	0.01852	0.0052
LP7 score	0	0	0	0	4.06775	4.65459
Length	0	0	0	0	−2.32699	−0.11491
Seqident	0	0	0	0	0.10018	0.12523
All-atom energy	−391.63953	−467.4407	−360.05569	−341.10541	44.21736	75.8012
TE13	−19.40881	−19.40881	−93.16228	−97.04403	−7.76352	3.88176
OPTM_r	13.79275	10.01814	8.45132	8.21393	0.47479	0.09496
OPTM_e	−79.19568	−94.47518	−112.25831	−117.08147	−13.25451	−0.07364
SEQG_z	5.97931	7.75652	8.26605	7.97139	0.71211	−0.19951
SEQG_r	−22.20124	−20.61544	−9.51482	−4.75741	−1.5858	0
SEQG_e	373.74741	436.03867	488.98618	501.44446	46.71843	0
SEQL_z	−8.72111	−0.23882	14.64224	16.1987	−1.40822	−1.19411
SEQL_r	8.07139	29.05699	29.05699	30.67127	1.61428	0
SEQL_e	−238.48904	60.96713	509.25482	541.53156	8.96575	1.79315
TRDG_z	0.2669	0.2387	0.24801	0.23405	0.01425	−0.00029
TRDG_r	1.13958	1.9488	0.83891	0.72755	0.03341	0.00371
TRDG_e	966.18396	965.09589	748.57501	717.02167	22.84894	0
TRDL_z	−0.11305	−0.17718	−0.07919	−0.06786	−0.01223	0.00075
TRDL_r	−0.08091	0.25527	0.33048	0.37607	0.02963	−0.00228
TRDL_e	−3.07725	0.39608	11.94337	14.86828	0.42655	0
TRSG_z	−3.24856	−4.1038	−3.94917	−3.73981	−0.31473	−0.17789
TRSG_r	−9.96801	−8.77884	−4.65174	−3.9872	−0.20985	0
TRSG_e	−1159.66089	−1212.84448	−1002.70447	−961.19543	−27.24035	−1.29716
TRSL_z	1.98116	1.86288	−0.46326	−1.168	−0.69981	−0.82302

Table 1b

Feature	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
TRSL_r	−1.19125	−6.36468	−5.2415	−5.27554	−0.30632	0
TRSL_e	−4.25106	−11.71403	−18.94084	−22.53062	0.09447	0.85021
TSSG_z	0.17755	0.51506	2.08487	2.11651	−0.03867	−0.14239
TSSG_r	−3.4844	−4.30216	−6.22214	−6.29325	−0.49777	0
TSSG_e	−67.90762	−1.74122	52.81705	48.75419	−5.22366	0
TSSL_z	10.0814	10.29221	7.18801	6.82427	−0.2356	−0.92589
TSSL_r	−8.64243	−8.53092	−6.71879	−6.46788	−0.22303	−0.05576
TSSL_e	57.15934	57.39233	22.988	20.73579	3.88311	−0.07766
TSCG_r	−0.57523	5.03327	4.67375	4.88947	0.43142	0.03595
TSCG_e	29.48567	18.32893	73.05009	78.09719	4.78146	3.98455
TSCL_r	−5.55997	−10.18091	−5.73294	−5.58468	−0.74133	0.02471
TSCL_e	40.70287	33.57987	54.44009	55.45766	19.58826	20.09704
PSMG_z	−0.90546	−3.52235	−8.04966	−8.29841	−1.31839	−1.54725
PSMG_r	20.44518	33.22342	23.00083	20.44518	2.55565	0
PSMG_e	−261.13818	−395.54752	−579.88037	−618.28302	−7.68053	26.88187
PSML_z	81.79292	42.20973	16.31419	16.12922	−8.24958	−7.47271
PSML_r	−88.68544	−37.34124	42.00889	70.01482	2.33383	0
PSML_e	387.00385	−133.03256	−387.00385	−408.77277	−12.09387	67.72566
SRFG_z	0.35228	0.47261	0.18411	0.10655	−0.09568	−0.19209
SRFG_r	−0.26794	3.26883	1.12534	0.8574	0.05359	0
SRFG_e	72.91562	45.90983	−89.11909	−99.92139	−2.70058	−1.35029
SRFL_z	0.62809	0.84199	−0.05982	−0.1432	−0.35891	−0.31178
SRFL_r	4.90119	3.6114	8.10723	8.43889	0.84757	−0.03685
SRFL_e	8.87435	46.79204	29.31225	27.96765	6.18515	6.99191
TBLS_r	0	0	0	0	0	0
TBLS_e	−7.17571	−5.95059	−5.25052	−4.72547	−0.52505	0
TBSS_r	0	0	0	0	0	0
TBSS_e	−6.24964	−5.17827	−6.4282	−5.89251	−0.71424	0
SBLS_r	−0.05291	−0.00864	0.02268	0.00432	−0.00432	−0.00108
SBLS_e	17.09775	11.37199	1.19287	2.14716	0.39762	0
SSPOS	−0.05869	−0.06189	−0.06018	−0.06191	−0.00106	−0.00034

Table 1b

Feature	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
SSCOMP	0.05707	0.03722	0.03948	0.03266	−0.00099	0.00031
FREADY	0	0	0	0	0.00439	0.00035
SIFT	0	0	0	0	0.00566	0.00038
TM	0	0	0	0	0.00116	0.00046

*All-atom features obtained from the models generated by Modeller;

#Features not used in learning

Table 2

Cosines between branches 2-6 and final ranking

The overlaps between vectors of features of different branches using scalar products of normalized coefficient vectors. Note the significant similarity between branches 2 and 4. See text for more details.

	Branch2	Branch3	Branch4	Branch5	Branch6	Final Ranking
Branch2	1.0000	0.9677	0.8974	0.8782	0.2368	0.0954
Branch3	0.9677	1.0000	0.9557	0.9458	0.3245	0.0813
Branch4	0.8974	0.9557	1.0000	0.9976	0.4129	0.0305
Branch5	0.8782	0.9458	0.9976	1.0000	0.4247	0.0412
Branch6	0.2368	0.3245	0.4129	0.4247	1.0000	0.8838
Final Ranking	0.0954	0.0813	0.0305	0.0412	0.8838	1.0000

Table 3

Proteins / True pairs detected

Analysis of the recognition capacities of the second phase of LOOPP and of the individual branches. The analysis is reported for the Learning Set (LS), Test Set 1 (TS1), and Test Set 2 (TS2). For each set we report the number of proteins solved (at least one true pair is in the top 5) and the number of true pairs that we detected.

	LS		TS1		TS2	
	Proteins*	**True pairs**	**Proteins***	**True pairs**	**Proteins***	**True pairs**
Total	12527 (11694)#	209090	3779 (3505)#	39364	82	1125
Tree (total)	11420 (10795)#	162762	3568 (3353)#	30943	70 (54)#	768
Branch0	8269	62419	2594	12903	33	284
Branch1	5681	52727	1810	6798	30	224
Branch2	3602	22431	1066	4142	23	110
Branch3	2889	9592	795	1905	24	48
Branch4	2770	6947	763	1772	22	42
Branch5	2413	5177	628	1137	17	29
Branch6	2594	10226	721	2286	9	31

*Proteins with at least one hit present/identified.

#Given within brackets are the proteins with at least one T pair present/identified.

Table 4

Performance of Phase 2 tree

Evaluating the performance of the phase II of LOOPP with respect to the older version of LOOPP that was used in CASP7 (LP7)

	LS		TS1		TS2	
Performance Measure	**Phase 2**	**LP7**	**Phase 2**	**LP7**	**Phase 2**	**LP7**
The percentage of proteins where the best hit is in the top 5	90	70	93	75	91	70
The percentage of proteins where a true hit is in the top 5	94	89	93	89	80	68
The percentage of proteins where the best hit is the top hit	59	43	73	56	47	41
The percentage of proteins where the best hit is a true hit	90	84	90	85	74	55

Table 5

True and False hits in Top hits of LS

Analysis of true and false hits (a hit – a prediction by the server) for the Learning Set (LS). The TM score between the model and the native structure is computed for the highest ranked model and for the best structural model in the top 5. The cutoff of TM=0.65 is used to differentiate between good and less-than-good models.

	Top 1		**Best model in Top 5**	
	TM<0.65	**TM>=0.65**	**TM<0.65**	**TM>=0.65**
True (T)	733	9503	789	9835
False (D)	721	463	514	282
Total	1454	9966	1303	10117