Published in final edited form as:

Proteins. 2008 August 15; 72(3): 910–928. doi:10.1002/prot.21976.

A template-finding algorithm and a comprehensive benchmark for homology modeling of proteins

Brinda Kizhakke Vallat¹, Jaroslaw Pillardy², and Ron Elber³

¹Department of Computer Science, Cornell University, Upson Hall 4130, Ithaca, NY 14853

²Computational Biology Service Unit and the Cornell Theory Center, Cornell University, Ithaca, NY 14853

³Department of Chemistry and Biochemistry, Institute of Computational Engineering and Sciences, University of Texas at Austin, 1 University Station, ICES, C0200, The University of Texas at Austin, Austin TX 78712

Abstract

The first step in homology modeling is to identify a template protein for the target sequence. The template structure is used in later phases of the calculation to construct an atomically detailed model for the target. We have built from the Protein Data Bank a large-scale learning set that includes tens of millions of pair matches that can be either a true template or a false one. Discriminatory learning (learning from positive and negative examples) is employed to train a decision tree. Each branch of the tree is a mathematical programming model. The decision tree is tested on an independent set from PDB entries and on the sequences of CASP7. It provides significant enrichment of true templates (between 50-100 percent) when compared to PSI-BLAST. The model is further verified by building atomically detailed structures for each of the tentative true templates with modeller. The probability that a true match does not yield an acceptable structural model (within 6Å RMSD from the native structure), decays linearly as a function of the TM structural-alignment score.

Keywords

Protein structure prediction; LOOPP Server; Mathematical Programming; Machine Learning

I. Introduction

Modeling protein structures has advanced significantly in the last few years, as is evident by the results of blind prediction experiments such as CASP ^{1,2} and Livebench ³. A prime factor in the observed significant progress in determining structures from sequences has been the increase in the availability of templates (proteins with solved structures). Templates are used to model the structures of targets (protein sequences with unknown structures). Another factor has been advancements in computer algorithms to recognize protein patterns, and the implementation of these algorithms in computer servers. These servers facilitate the easy and wide use of these techniques ⁴. The present paper is about a new matching algorithm to identify templates for a target protein. Once tentative hits are identified (Phase I) the matches are forwarded for further evaluation and building atomically detailed models (phase II). This algorithm forms the core of the LOOPP server for predicting protein structures (<http://www.loopp.org>).

II. Training, testing, and data sets

II. 1 The learning set

The first step in homology modeling is the determination of structural scaffolds (templates) for the sequence with the unknown structure (target). This step influences significantly the accuracy of the process and deserves careful examination. We present an extensive learning set for template identification and a prediction system that was learned from the data. We prepare a large number of protein pairs, some which are structurally similar and some which are not, for learning a prediction model from positive and negative examples. The training set was selected from the available folds of the Protein Data Bank (PDB) ⁵ and of the domains of SCOP ⁶ 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. However, this set was too limited for our prediction scheme. Our procedure for identifying templates for targets uses signals from global and local alignments. Global alignment penalizes protein pairs that differ appreciably in the number of amino acids and inclines to match pairs of comparable lengths. To allow for better matching with protein fragments (domains), we supplement the set with representative folds from the complete SCOP hierarchy ⁶ giving a total of 13,825 targets and templates in the learning set, which we call LS. The total number of possible pairs is $13,825 \times 13,824 / 2 = 95,558,400$, a number of comparisons that is too expensive to compute in detail. Actually the number is even larger since the targets and templates are not necessarily symmetric.

To increase efficiency we restrict the calculations to “candidates” for matches. The matches that are discussed below are selected based on structural comparison only. Candidates are determined by exploiting the inclination to match proteins of similar lengths, and by removing self-hits of SCOP domains to their corresponding PDB structures. Let the shorter and longer lengths of a pair of proteins be l_S and l_L , respectively. Pairs are considered “candidates” if $l_S/l_L \geq 0.5$. The reduction provides a total of 71,824,926 candidate pairs that were further evaluated by the structural alignment program TM-align ⁷. The TM score (TM score cutoff ≥ 0.5) was used to determine if a pair of candidate proteins in LS is classified as a tentative T (a tentative true match) or D (a decoy match). Of the total number of candidates about 1.4 million were tentative T and 76 million were D. A list of all tentative T and D pairs of the learning set is available at <http://www.loopp.org>. Other similarity measures were computed for candidate matches, (based on sequence, contacts, secondary structure information, etc.) and used to train the prediction model. The large number of comparisons of structures and other protein characteristics clearly requires significant computational resources and was performed on hundreds of CPUs over a period of a few months.

Our final goal is to generate a reasonably accurate atomically detailed model. It is not obvious if the structurally similar templates identified by TM-align, which we call tentative T (TM-scores greater or equal 0.5 ⁸) are refinable to acceptable structural models by the means that we have at hand. We define an acceptable model to be within 6.0Å RMS distance from the native shape of the target for 80% or more of the chain of the shorter protein. A tentative T template that generates an acceptable model (see next paragraph) is labeled a true T, which for brevity we call T.

The final promotion (or demotion) of the tentative T is done as follows: The structural alignments of another program (CE ⁹) are used as input to the program Modeller ¹⁰⁻¹². Modeller generates an atomically detailed structure of the target based on the fold of the template and an alignment of the target with the template. Modeller is a widely used

resource in the field, and is a component in our structure prediction system LOOPP <http://www.loopp.org>. It therefore is a natural choice for this task. The input of structural alignments to Modeller ensures that a high quality alignment is provided and that the errors in the final product are primarily the choice of the template and the model building capacity of Modeller (and not the alignment). We address the problem of finding score parameters that yield optimal alignments elsewhere^{13,14}. These parameter sets are based only on sequence information for the target and the alignments they provide thrive to mimic structural alignments. Here we focus on other aspects of the problem.

Why do we not use both CE and TM-align to generate the structural alignments for Modeller? The problem is cost. It would take a few more months of CPU on 100 processors to build atomically detailed structures for all tentative T in LS with the two structural alignments, from CE and from TM-align. We need to make a practical choice of using only one of them. While TM-align is significantly faster than CE, which is important for our large-scale computer experiments, we have considerable past experience with modeling based on the local alignments of CE⁹. Therefore, we model LS structures using CE alignments of tentative T pairs detected by (much faster) TM-align. Nevertheless, comparing model accuracy generated by different structural alignments is of considerable interest. Therefore in one of the test sets (TS1 see next section), which is much smaller in size than LS, we try them both. CE alignments provided 77,978 true hits while tm alignment 65,761. The merged set provides 91,445 T. While CE shows slightly better performance, the two algorithms are clearly complementary. It seems beneficial to use more than one optimal alignment (even in the context of structural alignments) to generate a corresponding model.

Of the proteins we examined in LS 1,136 did not have any true hit (but all proteins had at least one tentative T pair with TM score higher or equal to 0.5, suggested as a hit in reference⁸). The distribution of tentative T that did not make it to true T is shown in figure 1. It is a linearly decreasing function of the TM score in the range of TM between 0.5 and 0.8. It provides consistently and steady high performance between 0.8 and 1. We re-classify tentative T that did not produce acceptable models as D (decoys) for the purpose of learning. The number of T templates for a particular target varies from protein to protein with a maximum of 361 (figure 2). This set of T and D pairs (the number of remaining T and D are 418,037 and 71,406,889 respectively) was used to train a prediction model. The prediction model was evaluated on the test sets (TS) discussed below.

II. 2 Test sets

The tests match a template in LS with a target from a new collection of proteins. It is important to design test sets (TS) that are independent of the learning set (LS) to avoid “over-learning”. Over-learning is usually referred to the design of a complex model, which is highly accurate on the training set, but is not performing similarly on other sets with similar properties. Ideally the classification of T and D pairs on the TS should be as good as on the 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. The proteins are screened based on sequence similarity following the same procedure that was used for the LS filtering. We remove a TS1 protein when more than 70 percent of its sequence is identical to another protein in TS1 or in LS. This filtering provides us with a total of 4,183 proteins that are not present in LS. Then we compare the structural similarity of the TS1 folds with structures in LS and classify all TS1 – LS pairs to D and T pairs. The numbers of T and D pairs for TS1 were 91,445 and 22,004,925 respectively. Of the total number of proteins 381 out of 4,183 did not have any T hits, thus leaving 3,802 proteins in the TS1 dataset.

The second test set (TS2) is based on CASP7 targets (<http://predictioncenter.org/casp7/>). We evaluate all CASP7 targets similarly to the way we evaluated the structures in TS1 giving us a total of 1,691 T and 701,137 D pairs when matched to the LS. Nine out of 91 CASP7 structures were without a template according to our criteria, thus leaving 82 proteins in the TS2 dataset.

III. A system for template identification: Characteristics, Features, Similarity Measures, and Scores

III.1 The scores

The learning set is prepared according to structural alignments. A structural alignment requires the three-dimensional coordinates of the template and of the target. Obviously in real life applications we will not have significant structural information for the target protein, and the computational model to match a template to the target must use only sequence information of the target. In contrast a more complete description of the template (that includes structural information) is possible. Accordingly 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. On the target side the characteristic vectors are based on sequence only. However, we use the sequence in more ways than just the positions and identities of the amino acids. The sequence of the target is used to predict or compute properties that are matched with template characteristics:

- (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 acids of the target ^{15,16}.
- (iii) The SABLE prediction of solvent-exposed surface-area from the amino acid sequence ^{15,16}.

The actual sequence and predicted characteristics of the target are matched with the characteristics of the template, namely

- (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 ¹⁷).
- (iii) The actual exposed surface area of the template protein (extracted with the DSSP program, ¹⁷).
- (iv) THOM2 contacts between structural sites (used in threading calculations) ¹⁸.

There are many ways of combining and comparing these characteristics 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), match predicted secondary structure of the target with actual secondary structure of the template, etc. In the Appendix we provide a detailed list of the features that we have used in the present study. Note that in the proposed learning scheme there is no

need for the features to be unrelated. Some of the features we have used are strongly correlated.

Given two characteristics C_1 and C_2 , there is a need for an alignment to generate a score, a scalar feature value F_{12} . Alternative alignments and calculations of scores of the same pair of characteristics can differ appreciably in accuracy and speed, generating new features. In some cases speed is highly desirable and in others the highest confidence level is needed. In the modeling system LOOPP <http://www.loopp.org> we consider approximate alignments (BLAST), and exact (local or global) alignments determined by dynamic programming. We also use constant or structurally dependent gap penalties¹³.

Types of scores that we use are

- (i) Raw scores. For input characteristics C_1 and C_2 the raw score is $S(\bar{C}_1, \bar{C}_2)$ where $\bar{C}$ 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
- (ii) "Reverse score"¹⁹ is computed as $S(\bar{C}_1, \bar{C}_{2r}) - S(\bar{C}_1, \bar{C}_2)$ where C_2 is the reverse characteristic input of the second protein. For example, if C_2 is the protein amino acid sequence $C_2 \equiv a_1 a_2 \dots a_n$ then $C_{2r} \equiv a_n a_{n-1} \dots a_1$. The reverse sequence provides an inexpensive measure of the deviation of the raw score from a match by chance. It is particularly effective in reducing the scores of low complexity sequences (sequences with many repeats or with a small number of amino acid types) that otherwise may score artificially high. Higher values of the reverse score suggest a better match.
- (iii) The E-values from variants of BLAST and PSI-BLAST²⁰.
- (iv) The Z score is the most accurate (and expensive) measure that we use to examine the deviation of true matches from matches by chance.

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

where the average $\langle S(\bar{C}_1, \bar{P}\bar{C}_2) \rangle_P$ is over the optimal scores of the permutations of the characteristic C_2 at the amino acid sites. A single permutation is denoted by $\bar{P}\bar{C}_2$. To obtain averages of sufficient accuracy we usually require hundreds to thousands of permuted sequences making the calculation of the Z score hundreds to thousands times more expensive compared to a single alignment of a sequence against another sequence. It is therefore useful to divide the process of structural modeling of proteins into phases and in the first phase execute only calculations that are inexpensive and can be performed quickly. The inexpensive first-phase calculation makes it possible to examine a large number of pairs in a reasonable time and propose candidates for further analysis. The present manuscript is focused on that phase (phase I). High scoring pairs of the first phase are forwarded to the second phase in which more accurate measures of similarity are computed and atomically detailed models are constructed. The requirement of rapid calculation in the first phase suggests that we will use only the raw and the reverse scores, and the E-value to compute features, and not the Z score.

III.2 Combining features to a single score

Given a set of features that measures the similarity between pairs of proteins (we have 40 features, some of them strongly correlated – see Appendix), we seek a computational model that uses these features to classify the pairs according to T and D. In machine learning there are numerous ways of combining multiple signals for classification and ranking. For example, the use of neural networks has been popular ²¹. However, the tools of Mathematical Programming (MP) ²² are also effective and suggest algorithmic clarity that is missing in the application of neural networks. MP is used in the present manuscript. We seek a score, Q , to rank the pairs, which is a linear combination of similarity measures, $P(F_1, \dots, F_k)$. The similarity measures are functions of the K features.

$$Q = \sum_{\alpha=1, \dots, K} \gamma_{\alpha} P_{\alpha} \quad (1)$$

The γ_{α} are the unknown coefficients to be determined from the learning set with MP. In the simplest implementation $P = F$ and we have a representation of Q as a linear combination of the F :

$$Q = \sum_{\alpha=1}^K \gamma_{\alpha} P_{\alpha} = \sum_{\alpha=1}^K \gamma_{\alpha} F_{\alpha} \quad (2)$$

However more complex relationships between P and F can be utilized. For example, the original values of F_{α} can be transformed to F_{α}^U such that the probability-density function of $P_{\alpha} \equiv F_{\alpha}^U$ is uniform between zero and one. These features seem to provide additional and complementary prediction capacity compared to straightforward F_{α} . Another transformation that we used is quadratic $Q = \sum_{\alpha} \gamma_{\alpha} P_{\alpha} = \sum_{\alpha, \beta} b_{\alpha\beta} F_{\alpha}^U F_{\beta}^U + \sum_{\alpha} a_{\alpha} F_{\alpha}^U$.

The quadratic transformation explores correlations between the features, but is still linear in the unknown coefficients $b_{\alpha\beta}$ and a_{α} . It is therefore accessible to Mathematical Programming learning approaches.

The training process requires similarity measures, P , and sets of D and T 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 $Q_{Tij} - Q_{Dik} > 0$ for a single comparison of a decoy (D) with a true match (T). This inequality is linear and the set of all inequalities that we consider can be written compactly as

$$\sum_{\alpha=1}^K \gamma_{\alpha} (P_{ij, \alpha}^T - P_{ik, \alpha}^D) > 0 \quad \forall ij, ik \quad (3)$$

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. The scalar product must be positive. If a solution exists then $\lambda \cdot \gamma$, where λ is an arbitrary positive scalar, is also a solution. To set a scale for the values of the parameters, and to avoid a trivial solution ($\gamma = 0$) it is convenient to write

$$\sum_{\alpha=1}^K \gamma_{\alpha} (P_{ij,\alpha}^T - P_{ik,\alpha}^D) > 1 \quad \forall ij, ik \quad (4)$$

We do not optimize a function of the coefficients (e.g. $\min[A\gamma + \gamma^T B\gamma]$ where A is a vector of length K and B is a $K \times K$ positive matrix) when solving (4), which leaves a volume of solutions to choose from (if the problem is feasible). However, with no function to minimize, the interior point algorithm²⁵ (which we use) finds a solution at the analytical center of the (bound) feasible volume²³. Centering the solution is useful for future predictions since it maximizes the distance of γ to coefficient vectors that violate the inequalities. Minimizing the norm of the coefficient vector (SVM²⁴ minimizes the norm 2 of a vector) is another approach to centering the solution. We have found that the interior point centering is slightly better than the norm 2 of SVM for the problems we have dealt with.

Equation (4) 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. It is, however, computationally expensive and requires significant computer resources to allow consideration of the vast number of inequalities.

The learning of the model is based on a relatively small number of feature types, a large number of examples (feature inequalities), and open-ended choices of the similarity measures. The similarity measures can be any reasonable function of the features. However, the highly flexible choice of the similarity measures is not without restrictions. It is unlikely that a very small change in a particular feature will point to a protein of a completely different class. Hence a similarity measure is expected to be a smooth function. It is also expected to have a similar performance (or to generalize well) on the test cases compared to the training set. After significant trials of different functional forms of the similarity measures we were not able to find a single ideal Q 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. 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. (4). An alternative formulation is required which is a slight adjustment of Eq. (4)

$$\begin{aligned} \sum_{\alpha=1}^K \gamma_{\alpha} (P_{ij,\alpha}^T - P_{ik,\alpha}^D) &> 1 - \eta_{ij,ik} \quad \forall ij, ik \\ \eta_{ij,ik} &\geq 0 \\ \text{Subject to } \min &\left(\sum_{ij,ik} \eta_{ij,ik} \right) \end{aligned} \quad (5)$$

where the $\eta_{ij,ik}$ are (positive) slack variables that make the solution of the new set of inequalities feasible. Formula (5) does not include centering (e.g. by minimizing the norm of the parameter vector like in SVM). For the problems at hand, which are highly infeasible, the model with centering was not better than the model without and we finally use Eq. (5) directly.

The PF3 code, which is the Mathematical Programming (MP) solver that we use²², is tuned specifically to problems like Eq. (5) frequently encountered in the field of bioinformatics (but not in the field of Mathematical Programming). In typical MP applications (as targeted in commercial packages), we find that the number of unknown parameters K is comparable

to the number of the weakly coupled constraints - N . This set up leads to a calculation of a sparse matrix of dimension $K \times N$. Special attention is given to efficient storing and manipulation of the sparse matrix by the use of pointers.

However, our problem is different. Equation (5) is unusual since the number of model parameters K is much smaller than the number of constraints (the inequalities) N , and the $K \times N$ matrix is highly asymmetric and dense. The program PF3²², which is based on the interior point algorithm, exploits this asymmetry for efficient, and massively parallel calculation of Eq. (5).

The actual number of pair comparisons (the number of inequalities) that we attempt to satisfy is staggering (1,764,718,884), but still significantly smaller than the national budget deficit. This is even after the application of the condition of length matching. One consequence of the large number of inequalities is that we are unable to consider all of them in a single run of the program PF3. We therefore learn the set of inequalities by sampling. For each protein we sampled 1,000 D pairs and up to 10 T pairs (if ten are available; in some cases the number is smaller than 10 (see figure 2)). With the number of proteins that we consider (12,689 proteins with at least one true hit), “only” 70 million inequalities are obtained using this sampling. The 70 million inequalities are loaded in our first cycle of attempting to solve the problem. PF3 provides a solution to this problem in a few minutes using 100 computer nodes. The solution was then checked for 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. Such a measure can be obtained from statistical theory (like in BLAST²⁶) which we do not have for our multiple signals, or from numerical studies of a large number of examples. Since our learning set is indeed large we can estimate the frequencies of T and D values quite accurately by binning empirical observations. A score is considered significant if the probability that it is a D ($\text{Pr}(D)$) is less than or equal to 0.001. Such a cutoff value generates a significant number of candidates enriched with T pairs that is still small enough to be analyzed in details in phase II. We define the significance score as $SC \equiv 1 - \text{Pr}(D)$.

In the version of LOOPP that participated in CASP7 as a prediction-server (called here LP7), a single linear combination of features, assisted by initial screening by BLAST, was used to determine the score. However, the assessment in the present manuscript suggests that a single linear score is insufficient (in figure 3 we showed the ranking accuracy of different models on the learning and training sets, including LP7). LP7 is using a single Q score in the classification of T and D and it is doing significantly better than using one feature (TBLS_e $\equiv$ PSI BLAST; see appendix) “as is”. Nevertheless, there remains a large number of T undetected. This observation suggests that a significant algorithmic revision of Phase I is warranted which is discussed below.

III.3 A tree of Q scores for template identification

The task of this section is to improve on a single Q prediction system that is obtained from MP learning. The proposal is to use multiple MP scores where different Q are learned in sequence on a shrinking template database. Consider the case in which some features carry extremely strong signals for a subset of the pairs. In this case the MP learning algorithm of a single Q score will preferably learn these pairs. It will not be easy to “convince” it to learn pairs with scores that are not so strong, but nevertheless significant. An extreme example of this type is of sequence to sequence-profile matches versus sequence to structure signals. Pairs of proteins with reasonably high sequence similarities are usually matched well with an algorithm like PSI-BLAST with no need for additional signals. On the other hand when

sequence similarity is very weak the importance of the PSI-BLAST signal diminishes compared to other features and a better approach to determine similar pairs may be matching of sequences to structures (e.g. by threading ²⁷). If we remove sequences detected with one Q score, it is possible to learn another score that focuses on the weaker signal. These sequential searches can be represented on a simple tree structure (figure 4).

III.3.1 The zero branch

We start our learning (and “real-life” prediction) by the application of PSI-BLAST. While it is possible (and we have tried it) to learn instead a linear combination of features, the PSI-BLAST signals are usually very strong. It makes a dominant contribution in the first (zero) branch, and significant contribution to other branches as well. It is therefore convenient to filter them out first, gaining in the process a simple interpretation of the results, and comparison to widely used annotation techniques. This initial filtering has no significant impact on the prediction capacity of our system since the next branches discover the pairs that PSI-BLAST missed. We have not noticed a major change in the overall prediction capacity when starting with a different score. This is similar to spanning a high dimensional space with a set of orthonormal vectors. The direction of the first vector does not impact the success of the protocol in covering the space.

We generate a PSI-BLAST profile for the target sequence using the nr database (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 are forwarded to phase II. All pairs that are not hits, are forwarded to the next branch.

III.3.2 The first and second branches

With the pairs recognizable by straightforward PSI-BLAST removed, we seek in the first branch a prediction model linear with the features F_α described in the Appendix. The coefficients $\gamma_\alpha^{(1)}$ (for branch (1)) are determined from equation (5). As discussed earlier it is impossible to load all the inequalities to the memory. We therefore sampled 70 million inequalities for the first branch (using at most 10 T pairs and up to 1,000 D pairs for each of these T pairs). The number of T pairs, some of which were eliminated in the first branch, was reduced for the second branch in which only 60 million inequalities were loaded to the MP program PF3.

We select predicted T pairs for these branches following a similar procedure to what is used for PSI-BLAST. 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 (i.e., $SC \geq 0.999$). In figure 5 we show the probabilities densities of T and D for the first 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.

The training of the second branch was identical to the first branch in all respects except that the set of pairs for the learning was different. It includes the remaining D pairs from the zero and first branches. As we illustrate in the **Results** section the recognition capacity of the second branch is diminished significantly compared to the first branch. If we continue with the same set of similarity measures and attempt to create a third branch from the data, the number of hits above the threshold is not significant. It is likely that after the second branch

we exhausted the capacity of the simple similarity measures ($P = F$) parameterized with one vector γ .

III.3.3 The third branch

uses similarity measures derived from the features that are transformed into a uniform distribution. We set $P_\alpha \equiv F_\alpha^U$ and repeat the learning of coefficients (56M inequalities were used for the learning). The same protocol for forwarding predicted T pairs to phase II, and moving forward the predicted D pairs to the next branch was used. This set of variables was effective (recognized a significant number of T pairs) only for a single cycle, i.e. after removal of identified pairs from the set, learning with the same set of variables did not produce a model that predicts a significant number of new hits.

III.3.4 The fourth branch

uses a quadratic expansion of the uniformly distributed variables $P_\alpha \equiv F_\alpha^U$. The chosen functional form for the similarity measures, as outlined earlier is

$Q = \sum_{\alpha} \gamma_{\alpha} P_{\alpha} = \sum_{\alpha, \beta} b_{\alpha\beta} F_{\alpha}^U F_{\beta}^U + \sum_{\alpha} \alpha_{\alpha} F_{\alpha}^U$ and 53 million inequalities were loaded to the MP solver PF3 at this branch. To reduce the number of coefficients to be determined, only twelve uniformly distributed features were chosen from the list of 40 (Appendix). Empirically we have found that features that made significant contributions in the previous branches continue to make significant contribution in the following branches but with coefficients of alternate signs. The chosen (uniform) variables for the quadratic formula contribute significantly to branch 3 (see Appendix for a detailed description of the features): OPTM_r, OPTM_e, SEQG_e, TRDG_e, TRDL_e, TRSG_e, TSSG_e, TSCL_e, PSMG_e, SRFL_e, TBLS_r, TBLS_e. These variables were fed into the program PF3 for another cycle of training and identification of T pairs. This branch was quite effective in adding to the overall prediction capacity of the model.

III.3.5 Final ranking

The fourth branch was the last. Further trials to construct new similarity measures and branches did not detect a significant number of T pairs and did not generalize well to the test cases. Therefore we did not add new branches. At this stage we have a collection of predicted T pairs from the four branches. How do we rank the predicted T pairs that are pooled together from the different branches? We used clustering and the significance score, SC, for final sorting of the matches.

Clustering the TM scores of the suggestive hits of the target protein is also used in the ranking. We build on the observation (figure 2) that many proteins have multiple true hits, which are likely to be similar to each other, and therefore generate sizable clusters of structural similarity. For all the L suggestive hits of target i we construct a network model. For pairs of templates with tm scores larger than 0.65 we assign a value of 1 (a network edge). Pairs of suggestive hits with lower scores were assigned the value of 0. The matrix of zeroes and ones is analyzed to determine clusters²⁸. We expect that larger clusters of suggestive hits are more likely to be T and hence are ranked higher. The significance score behaves similarly for all the branches. Therefore we use this score to rank the hits within clusters.

IV. Results

IV.1 Structural alignments

In figure 2 we show the log of the number of tentative pairs (TM score > 0.5) that were detected in the training set by the TM-align program. The proteins are sorted in descending order of the number of structural neighbors. Following an early spike (a large number of neighbors for a small number of proteins), the plot is approximately linear suggesting an exponential dependence of the number of neighbors on the sorted protein index (number of neighbors $\propto e^{-\eta i}$ where η is a fitted constant, and i is the sorted protein index). The coefficient is not only a good fit over a wide range of the curves, it also varies only slightly between the curves (η : 0.005 for all curves where the smallest value is for t-BLAST (0.0041) and the largest value (0.0068) is for the curve of structural alignments with TM-score greater or equal 0.5)). While the curves are clearly exponential the coefficient is relatively small. If the curves are examined for a relatively small number of proteins (e.g. hundred), the exponent can be expanded, and appeared to be a power-law: $\exp(-\eta i)$; $1 - \eta i \approx 1$. On the same graph we also draw the number of T pairs for which Modeller generates accurate structures. As mentioned above, the curves of the predicted hits seem to have a similar exponential coefficient to what was observed for the structural alignment ($\eta i = 0.0056$); however the amplitudes are different. Modeller is successful in generating a fraction of the total number of neighbors; a log of fraction which seems roughly a constant throughout the majority of the dataset. The exceptions to the constant fraction are proteins with a very large number of structural neighbors on which Modeller is doing even better. The exponential behavior (as opposed to a power-law) suggests a more rapid decay in the number of available templates across protein space, which will make the search for homologous proteins at the tail of the distribution significantly more difficult.

The cutoff of the TM score of 0.5 may be too low for use with Modeller. While there is a strong correlation between the structural alignment score and the ability to generate an accurate model, the correlation is not perfect. In figure 1 we show the distribution of tentative T value of pairs that were unsuccessful in generating accurate structures. The distribution is maximal at the origin (for minimal values of TM scores that were still considered hits) however, the tail of the distribution lingered to TM values of (even) 0.9. In figure 6.1 we show an overlap of two structures with structural alignment score above threshold. While the overall secondary structure elements are predicted correctly, the packing of elements is quite different. As a result, and perhaps not surprisingly, Modeller failed on this example. The second example (figure 6.2) is of higher tm score (0.75) with significant agreement between the two shapes. Modeller was not able to generate an accurate model also from this template. The lengths of loops to be modeled in cases that Modeller failed were comparable to cases in which Modeller did not fail. We were not able to identify a clear source for the problems with Modeller. On a more positive note (there are many positive notes in our data) we show in figure 6.3, an application of Modeller to a non-trivial T pair for which a successful model was built (by non-trivial we mean that PSI-BLAST was unable to detect it). Hence, the last picture illustrates the added value of our learning compared to straightforward application of PSI-BLAST for T pair detection.

IV.2 Description of the tree

The computed coefficients for all 4 branches are summarized in table 1 and are also available from the web from <http://www.looppp.org>. For the purpose of comparison it is

convenient to normalize the vector of coefficients such that $\sum_{\alpha} \gamma_{\alpha}^{-2} = 1$ for all branches. The values of the normalized and un-normalized coefficients are provided for convenience. The normalization does not change the order of the ranking.

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 suggesting alternate combinations of similarity measures. It is of interest to examine how similar are the coefficients of the vectors. The scalar product of the normalized coefficient vectors is 0.2 for the first two coefficient vectors, which suggests low overlap. The product of the coefficient vectors of the second and third branches is particularly low (0.0587) while the first and the third product produces somewhat higher value (0.1246). Overall there is little similarity between the coefficient vectors of different branches.

Perhaps not surprisingly, given the experience gathered in the field, the dominant signals are based on sequence and profile comparisons. Nevertheless, as we move forward in the branches we find that the PSI-BLAST signals are getting weaker and other contributions become important. Consider the score of TBLS_e (the score used in branch zero). It has one of the largest coefficient in the normalized first branch (-0.3415) and it “decays” to -0.0142 in the third branch (the more negative is the energy, the signal is more positive). The third branch has a mixture of a number of features, some of them are sequence related, but some are not. The PSMG_e is a score of aligning a profile of the template against the sequence of the target. It is increasing between branches (-0.0181 in branch 1 to -0.3602 in branch 3). The mixture of sequence and threading TRSG_e has one of the highest amplitudes in the normalized branch 3 changing from 0.0325 on the first branch to -0.3403 in the third.

However, not all contributions are easy to interpret. The strong correlations of some of the features suggest that there are many cancellations. In the first branch, TBSS_e (PSI-BLAST variant, see Appendix for more details) has the largest coefficient (0.4667). However, TBLS_e makes a negative contribution (-0.3415). Since TBLS and TBSS are very similar and contribute in the opposite directions, the net result is a diminished contribution of the straightforward PSI-BLAST. In table 2 we show the relative correlation of sequence related features and others to the total score. The conclusion is that the collective contribution of all sequence-based signals is strongly correlated with the value of the total signal in the first branch but the correlation is weakened in the second and third branch to 0.53 and 0.60 respectively. The maximal correlation coefficient for other non-sequence features is a maximum of 0.36. Nevertheless, it is important to emphasize that the contribution of a subset of the sequence-based signals does not show such a strong correlation. Transitive BLAST (the PSI BLAST version that we used) has a negative correlation in the first branch and negligible correlation with the signal of branch 2 and 3. The contribution of OPTM_e (-0.1521) in the first branch is probably comparable to PSI-BLAST. The last feature (OPTM_e) is a mixture of threading, secondary structure, and sequence alignment substitution tables¹³.

Another example of a complex behavior of the MP model is the coefficient of the local threading energy (TRDL_e) which is positive in the third branch (0.2495). This suggests that better threading energies (which are negative) are lowering the overall score. Nevertheless, threading is making useful (positive) contributions even if somewhat indirectly. Other threading variants TRSL_e, and TSSL_e, which are mixed substitution tables that include threading, sequence alignment, and secondary structure information²⁹, are making significant positive contributions to the total score. These positive scores are balancing the negative score of pure threading. A similar observation is made also for the score of global threading alignment (TRDG_e). Overall, it seems that the mixed substitution tables carry more signal than the pure scores, in accord with earlier studies²⁹. Some features do not make significant contributions throughout the learning process (e.g. KMER and CFTR) and probably could be removed. The flexibility of the MP procedure allow us to evaluate many different features with essentially no performance penalty for including too many of them.

By the end of the day, it is not necessary to re-compute the model without these features since their contributions are small anyway.

The fourth branch has a different number of variables. Its coefficients are displayed in table 1.b (12 features in a linear and quadratic expansion of the score). TBLS_e and TBLS_r make the most significant positive contributions to the linear expansion term. In the quadratic form however the diagonal term of TBLS_r makes a significant negative contribution, while TBLS_e continues to make positive contribution. The contribution of TBLS_r to the overall recognition is non-linear and the quadratic and linear terms “compete”. TBLS shows little coupling to other features and its off-diagonal elements are much smaller than the diagonal elements. Another major “contributor” is global threading (TRDG_e) that makes negative contribution to the total score in both the linear and the (diagonal) quadratic term. In reverse to TBLS, TRDG_e has very significant coupling to other threading variants (TRSG, TSSG and TSCL) that, when coupled to TRDG_e, make positive contributions to the overall score.

IV.3 Evaluating performance

There are numerous ways of measuring the performance of target recognition. We have chosen three different measures to assess the prediction capacity of the tree, and of the individual branches. The first measure we considered is the number of T pairs that were detected. It is the most straightforward measure on which we also trained our model. The maximum number of T in LS is of 418,037. Of this maximum PSI-BLAST detected 133,790 (32%), the first branch 83,746 (20%), the second 12,477 (2.9%), the third 13,572 (3.2%), and the fourth 16460 (3.9%). The total number of T pairs solved by the tree is therefore 260,045 (62%) (data in table 3). In figure 3.1 we compare the ranking of T and D pairs that are detected by the tree, by LP7, and by PSI BLAST (the feature TBLS_e – see Appendix) with cutoff E-value of 1.9. The high E value was used since we were less concerned with the false hits (to be removed in Phase II) and focused instead on increasing the number of tentative hits in our set. Further increase in the PSI-BLAST cutoff E-value does not increase the number of T hits in the ranges considered.

While it is evident that the tree captures more T pairs, a few comments are in place. First, at the level of a single score PSI-BLAST is doing extremely well and the improvement provided by our complex scheme is significant but not profound. It is about a factor of two for the training set and the test sets (see below). Second, the tree structure also captures more false hits than PSI BLAST (figure 3.2). These extra false hits will need to be eliminated in phase II, depending on the resources available for the filtering. The tree output is more expensive to evaluate compared to the filtering of the PSI-BLAST output. Third, if we examine the beginning of the curves, then the tree model is the worst. It is the least successful in getting T pairs to the rank of the top 10 (figure 3.3 and 3.4). By enriching the number of T pairs in the first few hundred candidates we lose accuracy at the higher end of the prediction compared to LP7 and PSI-BLAST. This is perhaps another indication of the limit of the current features, gaining at one front comes at the cost of losing some accuracy in another application. If the computational resources to evaluate at least one hundred candidates in phase II are not available, then LP7 or PSI-BLAST are better choices.

The tree model we generated on the training set generalizes quite well on the test sets. It has comparable performance on the two sets and in particular on TS1 which we consider more typical and comprehensive. The TS1 has a total of 91,449 true hits, PSI-BLAST detected 24,628 (27%), the first branch 18,990 (21%), the second branch 1,509 (1.7%), the third branch 6,358 (7.0%), and the fourth branch 4,205 (4.6%). The total number of T pairs detected by the tree is 55,690 (61%) similar to the performance of the tree on the LS (62%) (table 3). The performance of the second branch is significantly weaker in LS1 compared to

LS. This poorer performance of the second branch in TS1 is compensated by slight increases of the performances of other branches. In figure 7 we compare the actual ranking of the detected T pairs as well as the false hits of the tree, LP7, and PSI-BLAST with cutoff for E-value 1.9. The same comments made about the ranking for the LS set can be made also here.

Finally TS2 set (CASP7 structures) was evaluated. The statistics are considerably poorer and the CASP competition may bias the sampling of structures in a subtle way (e.g. the crystallographers willing to hold to their publication and to contribute to CASP may be working on non uniform subsets of protein space). Indeed LS is more similar to TS1 than to TS2. Nevertheless, the qualitative observations we made earlier, and the overall performance still hold. The total number of possible T for TS2 is 1,691 while PSI-BLAST detected 572 (34%), The first branch added 356 (21%), the second 4 (0.24% !), the third 116 (6.9%) and the fourth 59 (3.5%). The complete tree predicted correctly 1107 T pairs or 65% (table 3). A concise summary of these results is provided in figure 8. The overall performance is therefore comparable to LS and TS1 even though significant variations in the performance of the individual branches are observed. A significant observation is the poor performance of the second branch suggesting perhaps over-learning. Note that the prediction capacity of the second branch was low even on the LS. The other branches performed comparably to LS and TS1.

The number of hits is a useful measure. However, depending on the set of examples at hand it may be misleading. Consider the task of modeling two proteins. One of the proteins has 100 T templates and the second has only one. If our algorithm detects the 100 T pair of the first protein but not the true template of the second its performance is excellent according to the measure of the number of hits ($100/101=0.99$). However, the task of modeling the two proteins is only half done. There are two measures we propose to probe the performance on particular proteins. The first, which is demanding, is the number of proteins for which all T pairs were detected. A protein with all T pair detected is called “solved”. The statistics is summarized in table 3. From 12,689 proteins of the LS only 4,009 (32%) were solved. Of the T pairs detected 2318 (18%) were detected by PSI-BLAST and 1691 (13%) by the rest of the tree. The percentage are lower than the percentage of hits suggesting that there are a few “easy” proteins with a large number of T pairs that contribute to the high performance of the first measure. This is also confirmed in figure 2 that shows high success rates for proteins with a large number of T pairs. The performance on the test cases is consistent with LS. Only 19% of the proteins in TS1 and 23% in CASP7 are completely solved by the tree.

The measure mentioned above is probably too strict. When we model a protein we rarely need or are able to put to good use all true hits that were identified. A single (good) template per protein may do in many cases. A relaxed measure of performance of providing a model to a particular protein is therefore the number of proteins with at least one hit, which is the second measure that we use to evaluate our performance on particular proteins. We summarize these results in table 4. In brief, for 94% of the proteins in LS the tree identified at least one true template. In fact PSI-BLAST alone is doing quite well on the LS and is able to assign one template to 81% of the proteins of the LS. The rest of the branches of the tree beefed up the prediction by adding 12% new proteins with one hit. Most of the additions of hits by the rest of the tree are of T pairs for which PSI-BLAST already detected one (or more) true templates. Similar upbeat observations are seen in TS1 (the tree predicts one true hit for 95% of the proteins, PSI-BLAST 90% of total of 3802 proteins) and for TS2 (83% percent of the proteins are assigned at least one hit by the tree, 67% by PSI-BLAST of total of 82 proteins).

IV.4 Comparison to other CASP7 servers

It is of interest to compare our template selection algorithm with related approaches³⁰⁻³³. The results of the comparisons are summarized in table V. We select ten targets from the fold recognition category of CASP7 to sample diverse performances of the tree (some successful matches, and some matches that were not successful). For a meaningful comparison the prediction methods that we compare to must (i) use only one template, as we do, (ii) provide prediction to all the 10 proteins of the selected set, (iii) perform reasonably well. For each of these servers we consider the template that generates their best model. We calculate the TM scores of the templates and the targets to evaluate the quality of the templates (the higher is the TM score the better is the template). These scores are reported in table 5. For comparison we also list the template-target TM scores of our algorithm that participated in CASP7 (LOOPP – called here LP7). The LP7 matches are phase II results of the earlier algorithm of LOOPP.

The scores of the new selection algorithm proposed in the present manuscript are somewhat more complex to evaluate. The tree was designed to be a rapid screening method of a large number of candidates that provides a maximum number of true hits in the first 200 matches. These matches are forwarded to phase II of selection in which more elaborate evaluation of the matches is performed. No attempt was made to make the true hits ranked first, or in the top 5. To make the comparison meaningful we included three entries. The first entry is the best-scoring template according to the current, phase-I tree. This is the ranking if phase II is not used. This particular score is worse on the average than most of the other groups that we compare to. However if phase II is perfect and places the best template of the 200 candidates as number one we would obtain the second entry, which is significantly better than the other groups. We are in the process of designing a phase II score. While the design is still incomplete the preliminary results (Vallat et al. work in progress) are sufficiently promising and encourage us to include them in the third entry. The preliminary results of phase II score are still better than the matches of the other groups. Note also that PCONS6 and FAMS-ACE are meta-server while LOOPP is not.

V. Final remarks

We have developed a decision tree model for matching a sequence with a protein template and tested it comprehensively on independent sets. All the tests were performed on proteins that were not available when the learning set was constructed. The new model provides significant enhancement in matching capacity compared to PSI-BLAST and the old LOOPP identifier when the group of top 100 is considered. However, no improvement is obtained for the group of top 10. This makes it necessary to pass through a second filtering phase (called here Phase II) which is a topic of on-going research. Our comprehensive learning and testing sets make it possible to critically review the state of the art of homology. At present we are able to build with high confidence acceptable models only from a limited range of nearby pairs. From our estimates it takes templates with TM alignment scores between 0.8 and 1.0 to the native structures to produce acceptable models with probability that exceeds 90 percent. Considerable work is required to take full advantage of more remote homologous templates that have failure rates as high as 80 percent for tentative hits with TM scores of 0.55.

Acknowledgments

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

Appendix I

We describe the 40 different features that were used in the learning process (a branch in the decision tree is a linear combination of similarity measures that are functions of the features discussed below). Features are properties of pairs of proteins and are constructed by an alignment and scoring of two characteristics of the individual proteins. Examples of protein characteristics are: the amino acid sequence, the exposed surface area at every site, secondary structure, a sequence profile, etc. Characteristics are two dimensional arrays. One dimension of the array is of the same length of the protein and provides the index of the amino acid site. The second dimension includes the site properties such as the amino acid type, secondary structure, exposed surface area, profile, etc. The complete list is (i) the identity of the amino acid, (ii) computed and predicted secondary structure, (iii) computed and predicted exposed surface area, (iv) the number of THOM2 type neighbors²⁷, and sequence profile generated by PSI-BLAST. Some of the characteristics, which are based on structure, are available only for the template. The scores of features are obtained by aligning the two characteristics either by dynamic programming³⁰ or by BLAST. Scores that are used in the present manuscript are either raw scores of the above three alignments or reverse scores (see text for definition of a reverse score). In the design of the global score we kept both the raw score and the reverse score. Empirically we found that while there is some cancellation in linear combinations of the two signals, the prediction capacity is significantly higher if both are kept.

Therefore, for each pair of characteristics of two proteins we have a pair of scores (raw and reverse). We used 20 pairs of characteristics producing a total of 40 features. Each feature is presented first by a four-letter code corresponding to the characteristics that are compared. For example SEQG denotes global alignment of the two sequences using the BLOSUM 60 substitution matrix. The four-letter code is followed by an extension of _e for a raw score (energy) or _r for a reverse score. All the features below, unless stated otherwise, are calculated against target sequences derived from ATOM section of their PDB files, i.e. parts of the sequence for which the structure is unknown are omitted. The complete list of the 20 pairs of characteristics are listed below

SEQG	Global sequence alignment score with dynamic programming and the BLOSUM 60 substitution matrix ³¹
SEQL	Local Smith-Waterman sequence alignment score with the BLOSUM 60 scoring matrix
OPTM	A local alignment computed with dynamics programming based on a substitution matrices learned from structural alignment and includes sequences, exposed surface area and secondary structure information ¹³ .
TRDG	Global alignment of sequence into structure (threading) based on the THOM2 energy function ²⁷
TRDL	Local alignment of sequence into structure (threading) based on the THOM2 energy function.
TRSG	The score of a global alignment of a mixed model (a linear combination of substitution matrices) which includes BLOSUM 60 (sequence against sequence) and THOM2 (sequence against structure) ²⁹).
TRSL	The same as TRSG but this time for local alignments.
TSSG	The score of a global alignment with a scoring model which is a linear combination of three substitution matrices; BLOSUM 60 for sequence alignment, THOM2 for threading, and the predicted secondary structure of SABLE ¹⁶ against calculated secondary structure for the template using the DSSP program ¹⁷ . The secondary structure substitution table is 1 if the predicted and actual secondary structures are the same and zero otherwise. Gap penalty was -1. The coefficients for mixing the matrices were optimized by Craig Lowe

	(unpublished) following the protocol described in ²⁹ .
TSSL	The same as TSSG, but this time for local alignment.
TSCG	The score of a global alignment of predicted secondary structure (by the program SABLE ¹⁶) against the actual secondary structure calculated from the structure by DSSP ¹⁷ . The substitution table assigned a value of 1 if the predicted and computed secondary structure of an amino acid are the same and zero otherwise. Gap penalty was -1. The substitution table was designed by Craig Lowe (unpublished).
TSCL	The same as TSCG but this time for local alignment.
PSMG	The score of a global alignment of the sequence of the target against a pre-computed profile of the template using PSI BLAST against nr. One iteration was used to generate the profile with a threshold of 0.001. The substitution matrix was BLOSUM 60.
PSML	The same as PSMG, this time for local alignment.
SRFG	The score of a global alignment of the exposed surface area of the target (predicted by SABLE) against the actual exposed surface area of the template (computed by DSSP). The substitution matrix was derived from a statistical potential.
SRFL	The same as SRFG, this time for local alignment.
TBLS	Transitive BLAST. Computed an alignment of the sequence of the template against a profile of the target with PSI-BLAST. A profile is computed for the target against NCBI nr database using PSI-BLAST with e-value threshold of 0.01 and 3 iterations.
KMER	(A study by Uri Keich (unpublished)). Employing k-mer statistics to detect similarity between sequences. A probability function is estimated for finding a particular fragment (a k-mer) of amino acids in a protein sequence. The k-mers differ in length from 1 to 10. The distributions so created for the target and the template are compared to each other to yield a similarity score. Unfortunately, this feature did not make a significant contribution to the overall score.
CFTR	(A study by Joon H. Han, unpublished) A score that correlates the probability of a spatial contact with the distance along the sequence. This feature provides little contribution to the overall score.
SBLs	Standard BLAST with full sequence of the target (SEQRES), including parts of the sequence for which the structure is unknown.
TBSS	The score of transitive BLAST computed as with TBLS but using the full sequence of the target (SEQRES) instead of the ATOM record. TBSS generates results almost identical to TLBS. Probably only one of them should be included.

References

1. 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]
2. Kryzhafovyeh A, Venclovas C, Fidelis K, Moult J. Progress over the first decade of CASP experiments. *Proteins-Structure Function and Bioinformatics* 2005;61:225–236.
3. Rychlewski L, Fischer D. LiveBench-8: The large-scale, continuous assessment of automated protein structure prediction. *Protein Science* 2005;14(1):240–245. [PubMed: 15608124]
4. Fischer D. Servers for protein structure prediction. *Current Opinion in Structural Biology* 2006;16(2):178–182. [PubMed: 16546376]
5. 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]
6. 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]
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. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. *Proteins-Structure Function and Bioinformatics* 2004;57(4):702–710.

9. Shindyalov IN, Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering* 1998;11(9):739–747. [PubMed: 9796821]
10. Sali A, Blundell TL. COMPARATIVE PROTEIN MODELING BY SATISFACTION OF SPATIAL RESTRAINTS. *Journal of Molecular Biology* 1993;234(3):779–815. [PubMed: 8254673]
11. Marti-Renom MA, Stuart AC, Fiser A, Sanchez R, Melo F, Sali A. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure* 2000;29:291–325.
12. 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]
13. 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.
14. Chun-Nam JY, Thorsten J, Elber R. Support Vector Training of Protein Alignment Models. *Lecture notes in bioinformatics: RECOMB 2007*. 2007
15. 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.
16. 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.
17. 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]
18. 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.
19. 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]
20. 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]
21. McGuffin LJ, Jones DT. Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics* 2003;19(7):874–881. [PubMed: 12724298]
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 University Press; Cambridge: 2000.
25. Wright, SJ. Primal-Dual Interior-Point Methods. SIAM; Philadelphia: 1997.
26. 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]
27. Meller, J.; Elber, R. Computational Methods for Protein Folding. Vol. 120. *Advances in Chemical Physics*; 2002. Protein recognition by sequence-to-structure fitness: Bridging efficiency and capacity of threading models; p. 77-130.
28. West, D. Introduction to graph theory. Prentice Hall; NJ: 1996.
29. Teodorescu O, Galor T, Pillardy J, Elber R. Enriching the sequence substitution matrix by structural information. *Proteins-Structure Function and Bioinformatics* 2004;54(1):41–48.
30. Mt-Skkmidtahkohu, Genki Terashi. Fams-ace: A combined method to select the best model after remodeling all server models. *Proteins: Structure, Function, and Bioinformatics* 2007;69(S8):98–107.

31. Zhou H, Zhou Y. SPARKS 2 and SP3 servers in CASP6. *Proteins* 2005;61(Suppl 7):152–156. [PubMed: 16187357]
32. Wu Y, Lu M, Chen M, Li J, Ma J. OPUS-Ca: A knowledge-based potential function requiring only C{alpha} positions. *Protein Sci* 2007;16(7):1449–1463. [PubMed: 17586777]
33. Wallner B, Elofsson A. Pcons5: combining consensus, structural evaluation and fold recognition scores. *Bioinformatics* 2005;21(23):4248–4254. [PubMed: 16204344]
34. Durbin, R.; Eddy, S.; Krogh, A.; Mitchison, G. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press; Cambridge: 1998.
35. Henikoff S, Henikoff JG. AMINO-ACID SUBSTITUTION MATRICES FROM PROTEIN BLOCKS. *Proceedings of the National Academy of Sciences of the United States of America* 1992;89(22):10915–10919. [PubMed: 1438297]

Figure 1. The fraction of suggestive T that failed to yield a successful structure ($\text{RMSD} < 6\text{\AA}$) with Modeller, as a function of the TM score. Data shown for (a) LS (b) TS1 (c) TS2.

Figure 2.

Top curve: The log of the number of structural neighbors to a protein based on TM score (≥ 0.5) in the LS. Second curve from top: Structural neighbors that were found to provide acceptable model with Modeller. Third curve from top: Found based on the tree designed in the present manuscript, Fourth curve: From LP7 and, fifth curve: From PSI-BLAST. See text for more details

Figure 3. Ranking accuracy on LS comparing the tree, PSI-BLAST (E=1.9), and LP7 (a) True pairs (T) in the first 200 suggestions (b) Decoys in the first 200 suggestions (c) True pairs (T) in the top 10 (d) Decoys in the top 10 (e) plot of true pairs Vs decoys for the complete range.

The E value was chosen particularly high to maximize the number of true hits (we care less in the present study with the number of decoys that will be filtered out in the next phase).

Figure 4.

(a) A sketch of the algorithm used to learn the decision tree (b) A sketch of the tree as it is applied to identifying templates for an unknown sequence (target).

Figure 5.
Overlap of the probability density of the scores for T and D pairs of the first branch of the LS.

Alignment of 153L and 1JSF (figure 6a)

Sequence Identity = 13%

RMSD = 2.8Å

Talign score = 0.59

```
DSSP      lllllllhhhl1llllllhhhhhhl1llllllhhhhhhhhhhhlHHHLLHHHHH.HHHHHH
153L      rtdcygnvnridttgascktakpeglsycgvsaskkiaerdlQAMDRYKTIK.KVGEKL    59
ident            |
1JSF      .....kvFERCELGARTLKrLGMDGY    20
DSSP      .....1LLHHHHHHHHHHhLLLLLL

DSSP      .LLLHHHHHHHHHHHLLl1ll1LLLLLL....LLEELLLEEELLLL.....
153L      .CVEPAVIAGIISRESHAGkvlknGWGDR....GNFGFLMQVDKRSHK.....    102
ident            |||
1JSF      rGISLANMCMCLAKWESGYN...tRATNYNagdrSTDYGIFQINSRYWCndgktpgavnac    77
DSSP      l1LLHHHHHHHHHHHLL..l1LEEEEl1ll1EEEELLLEEELLLLl1ll1ll1ll1ll1

DSSP      ..LLLLL.l1LHHHHHHHHHHHHhhhhhhhl1llllhhhhhhhhHHHHHHhhhh1ll1
153L      ...PQGTV..nGEVHTIQGTILINFIkti gkfpswtkdqqlkgGISAYNagagnvrsv    157
ident            |||
1JSF      hlsCSALLgdnlADAVACAKRVVRDPQ.....GIRAWV.....    110
DSSP      l1lHHHHHl1l1LHHHHHHHHHHHLLLL.....HHHHHLA.....

DSSP      l1ll1llhhhl1HHHHH.hhHHHHHHLL.l
153L      armdigtthddYANDVV.arAQYYKQH.G.y    185
ident            |
1JSF      .....aWRNCQnrdrVRQYVGCG.v    130
DSSP      .....hHHHHLLl1l1LHHHLLLLl1
```

d

Alignment of 1A05 and 1LMD (figure 6b)
Sequence Identity = 18%
RMSD = 2.7Å
Tmalign score = 0.75

DSSP	LLEEEEEELLHHHHHHHHH.HHHHHHHhhllLLEEEELLHHHHHHHLL	
1A05	MKKIAIFAGDGIPEIVAAR.QVLDAVDqashlGLCTEGLVGGAALDASDD	52
ident		
1LMD	adqrikvAKPVVMDGDENTRIIQFIKeKLLPHV....DVQLKYFDLGNRQDQND	55
DSSP	lllleellLLEEEEEELLHHHHHHHHHLLLE.....LLEEEEEELLHHHHHHHLL	
DSSP	LLLLHHHHHHHLLLEEEEEELLH..HLL...LLHHHhHHHHHHHHHLLLEEEEE	
1A05	PLPAASQLAMAADAVILGAVGSP..RWD...YPPAKRpeQGLLRKGLDIYANLRPA	107
ident		
1LMD	QVTIDSALATQKYSVAVKCATITPdeARVeEfKlKKMMK..SPMGTRIRILG.GTVFREP	112
DSSP	HHHHHHHHHHHLLLEEEELLHLLHhHHHhllLHLL...LHHHHHHHHL.LEEEE	
DSSP	ELLLLHhHLLHhHhHLLlEEEEEEELLHLLHLLLEEEEL.....	
1A05	QIFPQLLdaSFLRpeLVrdvDILVRELTDGIYFGQpRGLEVID.....	151
ident		
1LMD	ITCKNIP..RLVP..GWT.PITIGRHAHQGYKAT.DFYVDRAgfKivfepkdgasak	166
DSSP	LLLLLL..LLL..LLL.LLEEEEEELLHHHLE.EEEELLeeeeeeellllllle	
DSSP	LLEEEELLHLLHHHHHHHHHHHHHLLLEEEEEELLHLL.HHHHHHHH	
1A05	GRRGFTMYDEDEIRRIAHVAFRAQGRKQLCSVDKANVL.ETTLRWREV	203
ident		
1LMD	qwevynfPAGGVGMGMYNTDESISGFARSCFQYAIQRKFLYMSTKNTILKAYDRFKDI	226
DSSP	eeeeeeLLEEEELLHLLHHHHHHHHHHHHHLLLEEEELLHLLHLLHHHHHHH	
DSSP	HHHHH.HLL.....LLEEEHHHHHHHHHHHHHhLLEEEELHHHHHHHHHHHLL	
1A05	VTEVAR.DYFD.....VRLSHMYVONAAQLIRAPagFDVLLTGMMFGDILSDEASQIT	256
ident		
1LMD	FOEIFEKHYKTdfdkykIWEHRLIDOMVAQVLKSSG.GFVWACKNYDGDVQSDILAQGF	285
DSSP	HHHHHHHLLHHhhhlLLEEEHHHHHHHHHHHLL.LEEEEELHHHHHHHHHHHHH	
DSSP	LLHHLEEEEL..LLEEEELLHLLHhLL.....LHLLHHHHHHHHHHHHHLL	
1A05	GSIGMLFSASLG..EGRAMTEPIHGSAPIAG.....QKAAFLATILSVAMMLRHSLN	308
ident		
1LMD	GSGLMTSVLWpGdKTIEAaAGTVTRHYrehqgrPTSTNFIASIFAWTRGLEHARGK	345
DSSP	LLLLLEEEELlLLEEEELLHLLHHHHhhhlLHLLHLLHHHHHHHHHHHHH	
DSSP	LHHHHHHHHHHHHHLLHhHLL.....LHLL.....LHLLHHHHHHHHHLL	
1A05	AEFPAQVRVEAAVQRVLQGLRTADIA.....APGT....PVGTKAMAAVVA	353
ident		
1LMD	ldgnqGLIRFQAQLEKVCVETVESGAMTKDLAqihgLSNVkinhNFTSDFLOTIKSN	405
DSSP	hllhHHHHHHHHHHHHHHHLLHLLHHHHhhhlLHLLHLLHLLHHHHHHHHHHH	
DSSP	LLL....l	
1A05	LNL....k 357	
ident		
1LMD	LDReIgrq 413	
DSSP	HHHHhhll	

e

Alignment of 1A2YB and 1TVDA (figure 6c)
Sequence Identity = 22%
RMSD = 1.9Å
Tmalign score = 0.84

Alignment:

DSSP	lLEEEELL.LLELLLLLEEEEEEEELLlLLEEEEEEEELLHLLLEEEEEELLlLLEE	
Query	qVQLQESGP.GLVAPSQSLsITCTVSGfSLtCYGVNWVRQPPKGLEWLGMIWGdGNTDY	59
ident		
Sbjct	.DKVTQSSPdQTVASGSEVLLCTYDTVY.SNPDLFWYRIRPDYSFQFVFYGD.DSRSEG	57
DSSP	.LEEEELLLEEEELLLEEEELLLEELL.LLLEEEEEELLHLLLEEEEEELL.LLEE	
DSSP	LLLLHHHEEEEEHHHLEEEEEELLHhHHEEEEEEEEL...LLEeEELLLEEEEEEL	
Query	NSALKSRLSISKDNSKsQVFLKMNSLHTDDTARYCARER...DYRLdYWGQGTTLTVSS	116
ident		
Sbjct	ADFTQGRFSVKHILTKAFHlVISPVRTEDSATYYCAFTLpppTDKL.lFGKGRVTVVEP	116
DSSP	LHHHLLLEEEEEHHHLEEEEEELLHhHHEEEEEEEELlLHLLLE.EELLLEEEEEEL	

f

Figure 6.

(a) Structural superposition of 153L (Goose Lysozyme in blue) and 1JSF (Human Lysozyme in red) of fold: Lysozyme-like; (b) Structural superposition of 1A05 (3-Isopropylmalate dehydrogenase in blue) and 1LWD (Isocitrate dehydrogenase in red) of fold: Isocitrate / Isopropylmalate dehydrogenase-like; (c) Structural superposition of 1A2YB (Heavy chain of mouse monoclonal antibody of lysozyme in Red) and 1TVDA (T-cell receptor delta chain in Blue) of fold: Immunoglobulin-like; (d) alignment details for figure 6a where Modeller is unsuccessful at TM score 0.59; (e) alignment details for figure 6b where Modeller is unsuccessful at a high tm score (0.75); (f) alignment details for figure 6c where Modeller prediction is successful for a pair not detected by PSI-BLAST.

Figure 7.

Ranking accuracy on TS1. We compare the decision tree, PSI-BLAST (E=1.9) and LP7 (a) True pairs (T) in top 200 (b) Decoys in top 200 (c) True pairs (T) in top 10 (d) Decoys in top 10 (e) plot of true pairs Vs decoys complete range

Figure 8.

Ranking accuracy on TS2. We compare the decision tree, PSI-BLAST (E=1.9) and LP7 (a) True pairs (T) in top 200 (b) Decoys in top 200 (c) True pairs (T) in top 10 (d) Decoys in top 10 (e) plot of true pairs versus decoys for the complete range.

Table 1

The coefficient of the Mathematical Programming model: (a) The first three branches (b) Fourth branch with the quadratic terms. See appendix 1 for detailed description of the individual scores. The better raw scores (XX_e) are negative and the better reverse score (XX_r) are positive.

Coefficients learnt for the 40 features in the first three branches						
Feature	Branch1	Branch2	Branch3	Branch1 (Norm)	Branch2 (Norm)	Branch3 (Norm)
OPTM_r	-0.0025	-0.0288	0.1844	-0.0061	-0.0534	0.0298
OPTM_e	-0.0623	-0.0992	-2.3798	-0.1521	-0.184	-0.3844
SEQC_r	0.0037	0.0036	0.4571	0.009	0.0067	0.0738
SEQC_e	0.0054	0.0053	2.3574	0.0132	0.0098	0.3807
SEQL_r	0.0010	-0.0034	0.0315	0.0024	-0.0063	0.0051
SEQL_e	0.0031	-0.0003	0.4944	0.0076	-0.0006	0.0798
TRDG_r	-0.0082	-0.0169	-0.0074	-0.02	-0.0313	-0.0012
TRDG_e	-0.0102	-0.0095	2.6191	-0.0249	-0.0176	0.423
TRDL_r	0.0236	0.0192	0.1687	0.0576	0.0356	0.0272
TRDL_e	0.0444	0.0096	1.5448	0.1084	0.0178	0.2495
TRSG_r	0.0076	0.0178	-0.1111	0.0186	0.033	-0.0179
TRSG_e	0.0133	0.0066	-2.1073	0.0325	0.0122	-0.3403
TRSL_r	-0.0128	-0.0128	-0.1571	-0.0312	-0.0237	-0.0254
TRSL_e	-0.0409	-0.0002	-0.9787	-0.0998	-0.0004	-0.1581
TSSG_r	-0.0028	0.0032	-0.0679	-0.0068	0.0059	-0.011
TSSG_e	-0.0055	0.0062	-1.8221	-0.0134	0.0115	-0.2943
TSSL_r	0.0011	0.0064	0.0287	0.0027	0.0119	0.0046
TSSL_e	-0.0105	-0.0010	-0.6449	-0.0256	-0.0019	-0.1042
TSCG_r	0.0042	-0.0011	0.2075	0.0103	-0.002	0.0335
TSCG_e	-0.0003	-0.0045	0.3119	-0.0007	-0.0083	0.0504
TSCL_r	-0.0011	0.0023	0.0402	-0.0027	0.0043	0.0065
TSCL_e	0.0062	0.0089	0.2083	0.0151	0.0165	0.0336
PSMG_r	-0.0024	-0.0043	-0.2371	-0.0059	-0.008	-0.0383
PSMG_e	-0.0074	-0.0105	-2.2303	-0.0181	-0.0195	-0.3602
PSML_r	0.0042	0.0047	0.2523	0.0103	0.0087	0.0407
PSML_e	0.0061	0.0099	-0.0489	0.0149	0.0184	-0.0079

Coefficients learnt for the 40 features in the first three branches				
Feature	Branch1	Branch2	Branch3	Branch3 (Norm)
SRFG_r	0.0023	0.0043	0.1919	0.0056
SRFG_e	0.0022	0.0046	0.1398	0.0054
SRFL_r	0.0031	0.0129	0.2619	0.0076
SRFL_e	0.0039	0.0001	1.4297	0.0095
TBLS_r	-0.0733	0.2496	-0.2166	-0.1789
TBLS_e	-0.1399	0.2860	-0.0877	-0.3415
KMER_r	0.0000	0.0016	0.0000	0
KMER_e	0.0008	-0.0021	-0.3919	0.002
CFTR_r	-0.0002	-0.0010	-0.0501	-0.0005
CFTR_e	-0.0010	0.0000	-0.6478	-0.0024
SBLS_r	-0.1468	-0.2606	-0.0420	-0.3583
SBLS_e	-0.2217	-0.2552	-0.0420	-0.5411
TBSS_r	0.1630	0.0164	0.2440	0.3979
TBSS_e	0.1912	-0.0251	-0.1807	0.4667
				-0.0465
				-0.0292

Coefficients learnt in branch 4 for the 12 chosen features based on the quadratic form											
	0.0384	-3.3659	-3.3217	4.4551	1.4372	-4.3195	1.7305	-0.8026	3.1632	-0.2078	29.7989
											-55.9246
	OPTM_r	0.3654	0.3633	3.0147	-13.0092	-3.2127	-31.1677	0.2847	0.5252	-0.8874	-2.6794
	OPTM_e	0.3633	3.0147	-13.0092	-3.2127	-31.1677	0.2847	0.5252	-0.8874	-2.6794	-11.4066
	SEQG_e	-0.4817	-2.5808	2.3096	-3.2127	-31.1677	0.2847	0.5252	-0.8874	-2.6794	-15.6837
	TRDG_e	0.1697	-13.0092	-3.2127	-31.1677	0.2847	0.5252	-0.8874	-2.6794	-15.6837	-4.4900
	TRDL_e	0.1093	-0.1701	2.3676	0.2847	0.5252	-0.8874	-2.6794	-15.6837	-4.4900	1.3391
	TRSG_e	0.0949	12.2928	3.6303	5.7464	25.6331	14.5855	-2.2055	2.1515	2.8705	0.4614
	TSSG_e	0.1821	3.6303	5.7464	25.6331	14.5855	-2.2055	2.1515	2.8705	0.4614	-1.0182
	TSCl_e	-0.9092	-3.1021	0.1228	0.4885	0.1540	0.1540	0.1540	0.1540	0.1540	1.2491
	PSMG_e	-0.0365	0.1228	0.4885	0.1540	0.1540	0.1540	0.1540	0.1540	0.1540	-0.6386
	SRFL_e	0.0285	1.6748	3.9479	1.6748	3.9479	1.6748	3.9479	1.6748	3.9479	-0.3833
	TBLS_e	-0.0386	-0.0386	-0.0386	-0.0386	-0.0386	-0.0386	-0.0386	-0.0386	-0.0386	-0.0386

Coefficients learnt in branch 4 for the 12 chosen features based on the quadratic form													
	0.0384	-3.3659	-3.3217	4.4551	1.4372	-4.3195	1.7305	-0.8026	3.1632	-0.2078	29.7989	-55.9246	
	OPTM_r	OPTM_e	SEQC_e	TRDG_e	TRDL_e	TRSG_e	TSSG_e	TSCL_e	PSMG_e	SRFL_e	TBLS_r	TBLS_e	
TBLS_r	-0.0275	0.1540	0.8433	2.8705	-0.0594	-1.0055	-0.6386	-0.2277	-1.5970	-0.3833	-30.2543	-0.2567	
TBLS_e	-0.0386	-1.3295	1.0856	0.4614	-1.0182	1.2491	-1.1376	2.0910	-1.8326	-0.3425	-0.2567	55.0372	

Table 2
Correlation of contributions of different features (raw score and reverse score) with the total score in the LS

Correlation of the different sequence and non-sequence related coefficients of the Mathematical Programming model with the total score of the different linear branches.

Features	Branch1	Branch2	Branch3
TBLS+TBSS	−0.76	0.001	0.014
TBLS+TBSS+SBLs	0.64	0.021	0.015
TBLS+TBSS+SBLs+SEQG+SEQL	−0.36	−0.45	−0.17
TBLS+TBSS+SBLs+SEQG+SEQL+PSMG+PSML	0.83	0.53	0.60
Non-sequence based features (except CFTR, KMER, SRFG, SRFL)	0.26	0.30	0.36

Table 3

Proteins solved / True pairs detected

Summary of the number of proteins and the number of true pairs (Ts) that were solved in the sets LS/TS1/TS2, the overall performance and the contributions of the different branches.

	LS		TS1		TS2	
	Proteins	True pairs	Proteins	True pairs	Proteins	True pairs
Total	12689	418037	3802	91449	82	1691
t-Blast (1.9)	2771	171491	856	32770	11	727
LP7	2597	204704	766	37438	6	782
Tree (total)	4009	260045	1278	55690	19	1107
Branch0	2318	133790	712	24628	7	572
Branch1	1091	83746	338	18990	8	356
Branch2	60	12477	4	1509	0	4
Branch3	372	13572	143	6358	4	116
Branch4	168	16460	81	4205	0	59

Table 4
Proteins with at least one true pair solved

Number of proteins with at least one T pair solved in LS/TS1/TS2, overall performance and the contributions of individual branches.

	LS	TS1	TS2
Total	12689	3802	82
t-Blast (1.9)	11207	3410	56
LP7	11317	3408	55
Tree (total)	11878	3604	68
Branch0	10316	3160	46
Branch1	7488	2269	53
Branch2	2409	529	4
Branch3	2103	932	27
Branch4	3679	1669	20

Table 5

Comparison of template identification of Tree with other groups in CASP7: TMALIGN scores of the best template Comparing template-finding algorithms on 10 template and target matches from CASP7. The TM score is used to evaluate the quality of the match. The following servers are considered FAMS-ACE30, SP331, MA-OPUS32, SPARKS231, PCONS633, and LOOPP (as participated in CASP7). These servers are compared to the match of the tree that (i) is ranked first, (ii) the best match found in the first 100 matches (those matches are forwarded to phase 2), and to (iii) the best match found in a preliminary formulation of phase II (Vallat et al. work in progress).

Target/ Group	FAMS -ACE	SP3	MA- OPUS- SERVER	SPARKS 2	PCONS 6	LP7	TREE (Top 1)	TREE (Top 100)	Rank of the best template in phase 2 tree	Top template in phase 2 tree
T0290	0.97	0.87	0.47	0.87	0.97	0.58	0.95	0.97	1	0.97
T0293	0.61	0.61	0.61	0.61	0.61	0.61	0.65	0.75	2	0.61
T0303	0.88	0.88	0.65	0.86	0.88	0.88	0.72	0.8	1	0.8
T0322	0.8	0.8	0.72	0.8	0.8	0.8	0.8	0.8	1	0.8
T0331	0.5	0.48	0.5	0.27	0.51	0.58	0.5	0.62	3	0.5
T0340	0.78	0.88	0.88	0.88	0.88	0.88	0.63	0.88	21 [#]	0.78
T0362	0.85	0.85	0.85	0.85	0.85	0.84	0.78	0.84	1	0.84
T0372	0.6	0.67	0.67	0.67	0.67	0.67	0.58	0.72	NI [*]	0.66
T0378	0.76	0.76	0.72	0.76	0.76	0.76	0.72	0.84	NI [*]	0.76
T0381	0.65	0.65	0.65	0.65	0.65	0.65	0.86	0.91	1	0.91
Average	0.74	0.745	0.672	0.722	0.758	0.725	0.719	0.813		0.763

* : Best hit not identified. But a good true hit is identified.

: Although best hit is ranked 21, all 20 hits occurring above are reasonably good true hits.