SSALN: An alignment algorithm using structure-dependent substitution matrices and gap penalties learned from structurally aligned protein pairs

Jian Qiu and Ron Elber*

Department of Computer Science

Cornell University

4130 Upson Hall

Ithaca, NY 14853

*Corresponding author, e-mail ron@cs.cornell.edu, fax 607-255-4428

Keywords: alignment accuracy, secondary structure, relative solvent accessibility, position-specific gap penalties, structurally aligned protein pairs

Abstract

In template-based modeling of protein structures, the generation of the alignment between the target and the template is a critical step that significantly affects the accuracy of the final model. This paper proposes an alignment algorithm SSALN that learns substitution matrices and position-specific gap penalties from a database of structurally aligned protein pairs. In addition to the amino acid sequence information, secondary structure and solvent accessibility information of a position are used to derive substitution scores and position-specific gap penalties. In a test set of CASP5 targets, SSALN outperforms sequence alignment methods such as a Smith-Waterman algorithm with BLOSUM50 and PSI_BLAST. SSALN also generates better alignments than PSI_BLAST in the CASP6 test set. LOOPP server prediction based on an SSALN alignment is ranked the best for target T0280_1 in CASP6. SSALN is also compared with several threading methods and sequence alignment methods on the ProSup benchmark. SSALN has the highest alignment accuracy among the methods compared. On the Fischer's benchmark, SSALN performs better than CLUSTALW and GenTHREADER, and generates more alignments with accuracy > 50%, > 60% or > 70% than FUGUE, but fewer alignments with accuracy > 80% than FUGUE. All the supplemental materials can be found at http://www.cs.cornell.edu/~jianq/research.htm.

Introduction:

With the success of genome sequencing of many organisms, the database of protein sequences has been growing at an exceedingly fast speed. Although significant progress has been made in structural genomics, the growth rate of experimentally determined protein structures falls far behind that of protein sequences. Protein structure prediction with theoretical methods is of both intellectual interest and practical importance. According to CASP, protein targets can be classified into three categories: comparative modeling, fold recognition and new fold[1]. In comparative modeling, there exists a template in PDB sharing significant sequence similarity with the target such that it can be identified by sequence alignment methods. In fold recognition, there exists a template in PDB sharing significant structural similarity with the target, but the template cannot be identified by standard sequence-based methods such as PSI_BLAST. A method using structural information is needed to identify the correct template. In new fold (*ab initio*), the protein target possesses a new fold, and it does not share significant structural similarity with existing protein structures. Template-based methods cannot be applied to these targets. Instead, *ab initio* methods have to generate a large number of alternative structures not present in PDB and rank the structures correctly to find the closest structure.

For comparative modeling and fold recognition targets, a prediction model can be built based on the coordinates of the appropriate template. This approach generally involves three steps. In the first step, a search of a database of representative protein structures is performed to identify a good template that is structurally similar to the protein target. In the second step, an alignment between the target and the template is generated that should align structurally equivalent residues together as in the case of a structural alignment. In the third step, a prediction model of the target structure is constructed based on the alignment and the template structure. Although much progress has been made in template-based protein modeling, alignment errors remain a significant hindrance to the modeling accuracy[2]. Furthermore, alignment errors

introduced in step two generally can not be corrected in the final modeling step[3]. In principle, the first step and the second step can be combined in one alignment algorithm that attempts to find the best template and generate a good alignment at the same time. However, an algorithm good at the first step may not perform as well in the second step. In the identification of a template we have more flexibility than in seeking the best alignment. It is possible to use multiple scoring tables (e.g. for tertiary structure, secondary structure, sequence similarity, and more), and for each of the corresponding similarity measures to compute a (different) optimal alignment. The overall scores (each is a single number) are summed up to provide the global score for a match with a template structure. The use of multiple plausible optimal alignments and their corresponding scores are not possible to do with a single alignment (when we search for the optimal alignment). Therefore the search for the correct template with multiple alignments is more flexible, and has more opportunities to be optimized. In our experience with the suite of programs LOOPP http://cbsuapps.tc.cornell.edu/loopp.aspx we found that the SSLAN table (described in the present manuscript) has a significant contribution to the overall score (0.48) but could still use considerable help from other measures of similarity such as alignment using threading and exposed surface area. We therefore separate the two steps and focus on the second step in this study. We propose an alignment method that can reproduce the structural alignment with high accuracy given the knowledge of the correct template. *** expand on template selection **

Sequence alignment algorithms are able to generate good alignments when sequence similarity is relatively high. When sequence similarity drops into the twilight zone (less than 30% sequence identity), the problem becomes much harder and the performance of sequence alignment algorithms becomes much worse[4]. Since the goal is to reproduce the structural alignment accurately, inclusion of structural properties should help to achieve a better alignment. There are two types of energies that score the compatibility between a target sequence position and a template structure position: 1) Profile-type score[5; 6; 7] has the property that the score of

matching a certain sequence position type into a structure position is independent of the type matched into any other position. Standard dynamic programming algorithms such as Smith-Waterman[8] and Needleman-Wunch[9] can be used to efficiently generate an optimal alignment based on this kind of score. 2) Pairwise interaction energy[10; 11; 12] does not have this property and an optimal alignment can not be generated efficiently[13]. Stochastic search methods[14] or frozen environment approximations[15] have to be used during alignment generation. Because of this limitation of the second type, we develop a substitution matrix of the first type.

A number of methods have successfully used secondary structure and relative solvent accessibility information to improve the performance of protein fold recognition[16; 17; 18; 19; 20]. In a benchmark study of several alignment methods, Elofsson has shown that the inclusion of predicted secondary structure information improves the alignment accuracy[21]. Moreover, a number of studies focused on improving the quality of the alignment separately from template identification[22; 23]. The present work is in the same spirit as the studies of Zhang and Al-Lazikani. It focuses only on improving the alignment accuracy. In contrast to the previous approaches, no multiple sequence information (or multiple structure alignments) are explicitly used in our study. The accuracy of the alignment could have been increased further if multiple sequence alignment would have been used. Even without the use of profile alignment methods, we demonstrate that the current algorithm is highly accurate compared with leading alignment algorithms.

The essence of our algorithm is an optimized combination of scoring matrices based on information of amino acid type, secondary structure and exposed surface area matches. This is similar in spirit to a number of other approaches that were introduced in the past[6; 7; 18; 19; 23; 24; 25; 26; 27; 28]. Perhaps the two most important features of the current work are the following. All the score matrices (including the sequence substitution matrices) are learned from structural alignments, based on a single training set. Since our goal here is to be as similar as possible to structural alignments, learning a scoring matrix from a database of structural alignments is expected to be

advantageous compared with methods based on sequence comparisons, such as BLOSUM matrices[29]. The learning of structural alignments is similar to the work of Rice and Eisenberg, although the current work learns from a significantly larger training set and develops more detailed substitution matrices. The same training set is also used to learn structure-dependent gap penalties. While structure-dependent gap penalties were introduced in the past[3; 5; 19], here we studied the gap penalties in the framework of statistical potential with more detailed description.

In our approach, we develop a substitution scoring matrix with three components: 1) amino acid type substitution matrix; 2) target amino acid type *vs.* template secondary structure and solvent accessibility matrix; 3) predicted target secondary structure and solvent accessibility *vs.* template secondary structure and solvent accessibility matrix. A linear combination of these three component matrices is used to score a match of a target sequence position into a template structure position, similar to many other studies[6; 23; 24; 25; 26; 27; 28; 30].

Insertions and deletions are less likely to happen in the middle of an alpha helix or a beta strand, and occur infrequently in the hydrophobic core. Several groups have developed structure-dependent gap penalties and demonstrated their usefulness in improving alignment accuracy[19; 31]. In this study, we collect statistics of gap occurrences in structural alignments and derive statistics-based gap penalties dependent on the secondary structures and solvent accessibilities around the gap. The gap penalty at a position depends both on the position it is aligned to and on the positions before and after the gap.

We use a Smith-Waterman algorithm[8] with gap opening and gap extension penalties to generate an alignment between a target sequence and a template structure using the scoring matrix and structure-dependent gap penalties mentioned above. We test this alignment algorithm on the Prosup Benchmark, the Fischer benchmark, a set of CASP5 proteins and a set of CASP6 proteins. In this study, we show that this algorithm improves the alignment accuracy and does better in reproducing the structure alignments.

Methods:

I. Definition of structural environments based on secondary structure and solvent accessibility:

We use the program DSSP[32] to compute the secondary structure and solvent accessibility for each residue of a template structure. DSSP assigns each residue into one of eight secondary structure types: 3_{10}-helix (G), alpha-helix (H), pi-helix (I), beta bridge (B), extended beta sheet (E), bend (S), helix-turn (T) and other/loop (L). We combine type B with type E into one type E, and type S with type T into one type S, because they have similar amino acid preferences. The pi-helix type is very rare and has a distinct amino acid preference pattern that is different from all other types. Due to lack of enough statistics for this type, we arbitrarily assign pi-helix type into the type L (other/loop). Thus we classify each residue into 5 secondary structure types according to DSSP: G, H, E, S, L.

Many existing utilize the solvent accessibility information and classify an environment into two types: buried and exposed. However, some residues prefer to be partially buried instead of entirely buried or exposed (see Figure 2-1). Therefore, we classify each residue into 6 types according to its relative solvent accessibility X: 0: X=0, 1: 0 < X < 5%, 2: 5% <= X < 15%, 3: 15% <= X < 30%, 4: 30% <= X < 50%, and 5: X >= 50%. By combining these 6 types with the above 5 secondary structure types, each template residue is classified into one of 30 structure types according to its DSSP designation.

Several programs are available that successfully predict secondary structure[33; 34] and relative solvent accessibility for each residue of a protein from its amino acid sequence[35; 36; 37]. We use the program SABLE[37] to predict the secondary structure and relative solvent accessibility for each residue of a target sequence. SABLE was rated as having one of the best performances in secondary structure prediction in the recent CASP6 conference. SABLE predicts each residue as one of three secondary structure types: helix (H), strand (E) and other (C), and one of ten solvent accessibility types: 0-

9. We adopt the same secondary structure classification as SABLE, but combine the solvent accessibility types into 4 types: 0: 0, 1: 1, 2: 2-3, 3: 4-9. Combining these three secondary structure types and four solvent accessibility types, each residue of a target protein is classified into one of 12 structure types according to the SABLE prediction.

II. Substitution scoring matrix derived from structural alignments:

Since the goal is to generate an alignment close to the structural alignment between a target sequence and a template structure, we derive our substitution scoring matrix by learning from a database of structurally aligned protein pairs, as done in[18]. We derive substitution scores based on the preference of a sequence position type relative to the structure position type in the template to which it is aligned, rather than the preference of the sequence type of a position relative to the structure type of the same position in the protein's native structure. Given a position S of the target sequence and a position X of the template structure, we extract information such as amino acid type and SABLE predicted structure type from S, and information such as amino acid type and DSSP-assigned structure type from X and score according to their compatibility. The scoring matrix consists of three component substitution matrices: (1) amino acid type of S vs. DSSP type of X – matrix AD; (2) amino acid type of S vs. amino acid type of X – matrix AA (amino acid substitution matrix); (3) SABLE type of S vs. DSSP type of X – matrix SD.

For each structurally aligned position pair S and X (S from the target sequence, and X from the template structure), let (A_s, SA_s) represent the amino acid type and SABLE type of S, and (A_x, D_x) represent the amino acid type and DSSP type of X. The three component scoring matrices can be derived as follows:

(1) $$s(A_s, D_x) = -\log\left(\frac{P(A_s, D_x)}{P(A_s) \times P(D_x)}\right)$$

(2) $$s(A_s, A_x) = -\log\left(\frac{P(A_s, A_x)}{P(A_s) \times P(A_x)}\right)$$

(3) $$s(SA_s, D_x) = -\log\left(\frac{P(SA_s, D_x)}{P(SA_s) \times P(D_x)}\right)$$

A minus sign is introduced because our program minimizes the alignment score (in the analogy of an energy) instead of maximizing it. The score of a sequence position with types (A_s, SA_s) matching into a structure position with types (A_x, D_x) can then be expressed as $T=\alpha_1*s(A_s, D_x) + \alpha_2*s(A_s, A_x) + (1-\alpha_1-\alpha_2)*s(SA_s, D_x)$. α_1 and α_2 are two parameters that have to be determined empirically. There are only two independent parameters, since they are determined by ranking (i.e. which alignment is better than the rest) and the absolute value of the total score is not important in ranking.

III. Position-dependent gap penalties

Similarly position-dependent gap penalties can be learned from structurally aligned protein pairs. Given a particular environment type i for a gap, let N_{gi} be the number of times when a gap is in the environment of type i, N_g be the total number of occurrences of a gap in any environment type, N_i be the number of occurrences of environment type i, and N_t be the total number of occurrences of any environment type. The gap penalty g can then be derived as follows:

$$g(i) = -\log\left(\frac{P(i \mid gap)}{P(i)}\right) = -\log\left(\frac{N_{gi}/N_g}{N_i/N_t}\right)$$

The g derived from this formula can have negative values if a gap appears more frequently in a certain environment type than in the background distribution. To make sure that all the gap penalties are positive values (A positive value penalizes an alignment as the program minimizes the alignment score), a constant η is added to all the statistically derived gap scores.

Due to the asymmetry in protein threading, the insertion of a gap in the target sequence has to be treated differently from that in the template structure. When a gap appears in the target sequence, its environment is defined based on the structure

position X that it is aligned to, the last sequence position S_n before the gap, and the first sequence position S_{n+1} after the gap:

Target S_n $(-)_{0..j} - (-)_{0..k}$ S_{n+1}

Template X

The likelihood of a gap occurring in this environment depends both on the structure type of X, and on the predicted structure type of S_n and S_{n+1}. Let D_X be the DSSP-assigned structure type of X, and SA_n and SA_{n+1} be the SABLE-predicted structure types of S_n and S_{n+1} respectively. The gap penalty in the environment of (X, S_n, S_{n+1}) can then be expressed as $\beta_1*(g(D_X)+\eta) + (1-\beta_1)*(g(SA_n, SA_{n+1})+\eta)$.

Similarly when a gap appears in the template structure, its environment is defined based on the sequence position S that it is aligned to, the last structure position X_n before the gap, and the first structure position X_{n+1} after the gap:

Target S

Template X_n $(-)_{0..j} - (-)_{0..k}$ X_{n+1}

Let A_s be the amino acid type and SA_s be the SABLE-predicted structure type of S, and D_n and D_{n+1} be the DSSP-assigned structure types of X_n and X_{n+1} respectively. The gap penalty in the environment of (S, X_n, X_{n+1}) can then be expressed as $\gamma_1*(g(A_s)+\eta) + \gamma_2*(g(SA_s)+\eta) + (1-\gamma_1-\gamma_2)*(g(D_n,D_{n+1})+\eta)$. Because SABLE prediction may contain errors and SA_s may not reflect the real structure type of position S, $g(A_s)$ and $g(SA_s)$ are combined to compensate this problem.

It is well known that the distinction between gap opening penalty and gap extension penalty improves alignment accuracy. Separate statistics are collected for environments of opening gaps and extending gaps, and two separate sets of opening penalties and extension penalties are derived accordingly. Let g_o, g_e be the gap penalties computed according to the above scheme based on statistics of opening gaps and extending gaps respectively. The final adopted gap opening penalty is g_o and gap extension penalty is $0.1*g_e$. In other words, the gap extension penalty is on average one tenth of the gap opening penalty (This ratio is used the same as the one

recommended in [38], and is not optimized). The parameters β_1, γ_1, γ_2, and η are determined empirically (This is explained in detail in the next section).

IV. The training set

The training set is based on a representative data set developed previously in this group. It contains 1379 protein targets, and each target has a few associated template structures. The CE program[39] is used to generate the structural alignment between each target and template structure. Each target-template pair shares significant structural similarity with CE Z score larger than 4.5. The training set is divided into two parts. One set with 690 targets serves as the training set to learn the substitution matrices and the position-dependent gap penalties. The other set with 689 targets serves as the validation set to optimize the parameters, α_1, α_2, β_1, γ_1, γ_2 and η. For each sequence-structure pair in the validation set, an alignment is generated with the Smith-Waterman algorithm using the substitution matrices and gap penalties learned from the first set and a sampled choice of the parameters. The metric to be maximized is the total number of correctly aligned pairs compared with the CE structural alignment. The final choice of the parameters is: α_1=0.1, α_2=0.55, β_1=0.5, γ_1=0.1, γ_2=0.3 and η=1.5.

Results

I. The substitution matrices and the position-specific gap penalties

All the substitution matrices and position-specific gap penalties are available at http://www.cs.cornell.edu/~jianq/research.htm. Figure 1 shows the scores of three amino acids *vs.* DSSP types in matrix AD. Gly is more dependent on secondary structure than solvent accessibility with type S most favored. This is consistent with the high frequency of Gly occurring in turns. Ile, being hydrophobic, has a preference for fully buried environment (type 0), and also favors secondary structure type E (β strand). Tyr also prefers secondary structure type E, and achieves lowest energy in a partially buried environment. Figure 2 plots the scores of SABLE types H0 and E0

vs. DSSP types in matrix SD. As expected, Both H0 and E0 prefer DSSP solvent accessibility type 0. H0 favors DSSP secondary structure type H and disfavors secondary structure type E, while E0 favors DSSP secondary structure type E and disfavors secondary structure type H.

Figure 3 plots the scores of DSSP type-dependent gap opening penalties $g_o(D_x)$. A gap aligned to a buried structure position gets a higher penalty than that aligned to a more exposed position for all five secondary structure types. Secondary structure types G and S are most favored positions to be aligned to a gap while H and E are disfavored positions as expected. Interestingly, secondary structure type E is even less favored than type H for the same solvent accessibility type. This means that a structure position in a β-strand is less likely to be deleted than a position in a α-helix. All the gap penalty scores are larger than –1. With a default setting of η=1.5, all penalty scores become positive after addition of η.

For gap penalties $g(SA_n, SA_{n+1})$ and $g(D_n, D_{n+1})$, the number of solvent accessibility types for one position is decreased to two to allow enough statistics for each type. In $g(D_n, D_{n+1})$, type 0 has a relative solvent accessibility X < 15% and type 1 has X >= 15%. In $g(SA_n, SA_{n+1})$, type 0 corresponds to SABLE prediction type 0 and type 1 corresponds to SABLE prediction 1-9. Figure 4 plots gap penalties $g(SA_n, SA_{n+1})$ for secondary structure types E and C. As expected, a gap inserted in the middle of a β-strand (between two E positions) gets a higher penalty than at the boundary of a β-strand (between a E position and a C position). A gap inserted in the loop region (between two C positions) gets the lowest penalty. In addition, a gap inserted into a buried region (E0, E0) gets a higher penalty than that into an exposed region (E1, E1).

II. The CASP5 test set

This test set consists of 31 protein targets from CASP5 and up to ten templates for each target. These 31 CASP5 targets represent all the targets with the PDB code names available that we are able to find at least a template with CE Z score of at least

4.5. No domain divisions are attempted and the complete chains of the targets are used in the test set. Since the training set was developed before the CASP5 competition, no CASP5 target is present in the training set. Each template shares significant structural similarity with its target with a CE Z score of at least 4.5. There are 117 target-template pairs in total in the test set. The complete information of the targets and templates in this test set can be found at http://www.cs.cornell.edu/~jianq/research.htm. Three measures are used to evaluate the accuracy of the predicted alignments with respect to the CE structural alignments: 1) Nc: the total number of position pairs that are aligned identically between the predicted alignments and the structural alignments (number of correctly aligned positions); 2) Q: the average fraction of correctly aligned positions divided by the length of the CE structural alignment; 3) S: the average shift of positions in the aligned region.

Table 1 compares the performance of our method with PSI_BLAST[40] and SW_BLOSUM50. SW_BLOSUM50 implements a Smith-Waterman algorithm using Blosum50[29] and gap opening penalty of 10 and gap extension penalty of 1. PSI_BLAST was performed first against the NonRedundant database (NR) for five iterations to generate the position-specific scoring matrix (PSSM), and then against the templates in the CASP5 test set for one iteration to generate the alignments. In all three measures, our method is significantly better than both PSI_BLAST and SW_Blosum50 with a 47% improvement over PSI_BLAST and a 96% improvement over SW_Blosum50 in Nc, and a 48% improvement over PSI_BLAST and an 89% improvement over Blosum50 in Q.

To identify the contributions from component matrices and from the position-specific gap penalties, results are also computed for the substitution matrix AA, and the matrices AA+AD (no SABLE predictions), with the position-specific gap penalties and with a constant gap opening penalty of 1.5 and a constant gap extension penalty of 0.15. The structurally derived amino acid substitution matrix AA with constant gap penalties performs better than Blosum50, 11% improvement in both Nc

and Q. This demonstrates the usefulness of generating an amino acid substitution matrix based on structural alignments as observed in [41]. The current complete method (with all three matrices) performs much better than the matrices AA and AD only (no SABLE predictions). This indicates that the matrix SD possesses important information not captured by the other two matrices. The inclusion of matrix SD is especially useful at lowering the average shift from around 18 to around 6. By comparing the results of position-specific gap penalties with those of constant gap penalties, we find that the position-specific gap penalties are more useful when the matrix SD is not included, and only improves the performance slightly (2% in both Nc and Q) when all three matrices are used.

Figure 5 plots the fraction measure Q of SSALN, PSI_BLAST and SW_Blosum50 versus the percent sequence identity between the target and the template. All three methods perform better when sequence identity are high, with comparable performances in the region over 30% sequence identity. When sequence identity drops below 30%, SSALN and PSI_BLAST perform much better than SW_Blosum50. SSALN is also better than PSI_BLAST throughout this region and the improvement is more dramatic when sequence identity is below 20%. In quite a few cases, SSALN manages to generate reasonable alignments when sequence identity is low, while the other two methods fail to do so.

III. The ProSup Benchmark

This benchmark[42] contains 127 protein pairs with significant structural similarity but with sequence identity of no more than 30%. The structural alignments of the protein pairs are generated with the program ProSup[43]. The alignment accuracy is evaluated based on four metrics: 1) T_c: total number of correctly aligned residue pairs; 2) T_m: total number of missed residue pairs – number of residues in the template structures that are aligned in the structural alignments but not in the predicted alignments; 3) T_i: total number of incorrectly aligned residue pairs – number of residues in the template structures that are aligned in the predicted alignments but not

in the structural alignments; 4) σ_0: the average percentage of correctly aligned residues divided by the length of the structural alignment per protein pair.

Table 2 compares the performance of SSALN with several methods in Domingues *et al.*[42]. SSALN has the highest T_c and σ_0 and the lowest T_m and T_i among all the methods. In particular, SSALN shows a 20% improvement over the best threading method T_c (9256 *vs.* 7692) and a 21% improvement over the best threading method σ_0 (58.3% *vs.* 48.0%). Table 3 compares SSALN with PSI_BLAST[40], STROMA[44] and another threading method SPARKS[6] based on the σ_0 metric. SSALN is slightly better than SPARKS (the one with the incorporation of the single-body knowledge-based structure-derived score), and significantly better than the other methods. The correct alignments in the ProSup benchmark are derived from the structure comparison program ProSup, an independent program different from the CE program that is used to generate the training set. The evaluation of SSALN with an independent structural alignment program suggests that the performance improvement of SSALN is not sensitive to the structural alignment algorithm used.

IV. The Fischer's Benchmark

This benchmark[45] contains 68 protein pairs with significant structural similarity. Figure 6 compares the alignment accuracy of SSALN with methods CLUSTALW, FUGUE and GenTHREADER. The alignment accuracy is measured as the average fraction of correctly aligned positions divided by the length of the reference structural alignment. The alignments generated by the program COMPARER[46] are used as the reference as done in FUGUE[19]. SSALN has a higher number of alignments above all the five accuracy thresholds than both CLUSTALW and GenTHREADER. It performs better than FUGUE above the > 70% (20 over 14), > 60% (24 over 22) and > 50% (34 over 27) thresholds, the same as FUGUE above the > 90% threshold (3 in both cases), but worse above the > 80% threshold (7 compared with 10 in FUGUE). Some proteins in the Fischer's benchmark are present in the training set of SSALN.

To eliminate the effect of learning from the test set, a new training set is developed by removing protein pairs with either protein present in the Fischer set. This new training set has 1300 protein targets compared with 1379 in the original training set. SSALN (NoFischer) follows the same procedure as SSALN, but learns all the substitution matrices, gap penalties and parameters from this new training set. SSALN (NoFischer) gets very similar substitution matrices and gap penalties and the same optimal set of parameters compared with SSALN. Consequently, SSALN (NoFischer) has almost the same results as SSALN on the Fischer's benchmark, except that it has 6 alignments with accuracy higher than the > 80% threshold compared with 7 in the case of SSALN. This shows that the training set has enough statistics and the method is stable with regard to changes in the training set. Although all the training procedures are based on the CE structural alignments, SSALN is able to achieve good performance with an independent structural alignment method COMPARER.

V. The CASP6 test set

The CASP6 test set consists of 51 CASP6 targets with one suitable template for each target. We define a suitable template as a protein template that has a CE Z score of greater than 4.5, or a CE Z score of greater than 3.5 and RMSD of less than 3.5, when aligned against the target. The 51 targets in the CASP6 set are all the targets that we are able to find a suitable template. Among the 51 CASP6 targets, the LOOPP server (http://ser-loopp.tc.cornell.edu/cbsu/loopp.htm) is able to identify a suitable template for 30 targets. We call this 30-target subset the LOOPP_CASP6 set. The complete chains of all targets are considered with no attempts of domain division. The alignment accuracy is measured as the fraction of correctness -- the fraction of correctly aligned positions divided by the length of the CE structural alignment. When correctly aligned positions are defined as the positions that are aligned identically in both the predicted alignment and the reference structural alignment, the metric of fraction of correctness is called Q_0. We can also relax the standard of correctness by

allowing a shift of at most 4. When correctly aligned positions are defined as the positions aligned with at most a difference of 4 between the predicted alignment and the reference alignment, the metric is called Q_4.

Figure 7A and 7B show the alignment accuracy Q_0 *vs.* percent sequence identity of the protein pairs in the CASP6 set and in the LOOPP_CASP6 set for SSALN and PSI_BLAST. The PSI_BLAST alignments are generated similarly as in the CASP5 test. SSALN performs better than PSI_BLAST especially in the region of low sequence identity (<20%). For the CASP6 test set, the average Q_0 of SSALN is 63.4%, a 26% improvement over that of PSI_BLAST (50.1%). For the LOOPP_CASP6 test set, the average Q_0 of SSALN is 74.4%, a 14% improvement over that of PSI_BLAST (65.0%). When the more relaxed metric Q_4 is considered, SSALN displays a more dramatic improvement. For the CASP6 test set, the average Q_4 is for 82.9% for SSALN, compared with 56.2% for PSI_BLAST. For the LOOPP_CASP6 test set, the average Q_4 is for 89.1% for SSALN, compared with 70.8% for PSI_BLAST. This improvement can also be seen in Figure 7C and 7D, which show the alignment accuracy Q_4 *vs.* percent sequence identity of the protein pairs in the CASP6 and LOOPP_CASP6 set. Except for two protein targets in the CASP6 set, SSALN is able to generate an alignment with a Q_4 of greater than 50% for every other protein. PSI_BLAST, on the other hand, fails to do so for many targets in the low sequence identity region. It is especially promising that SSALN is able to align the majority of a protein approximately right (an error of at most 4 position difference) for most proteins.

VI. Prediction of CASP6 target T0280_1

In the recently completed CASP6, LOOPP server (http://ser-loopp.tc.cornell.edu/cbsu/loopp.htm) has the best prediction for domain T0280_1 according to the metrics GDT_TS, EQV4_0 and AL0 (http://predictioncenter2.llnl.gov/casp/casp6/public/cgi-bin/results.cgi). After identification of a suitable template, LOOPP server uses SSALN to generate the

alignment between the template and the target. The program MODELLER[3; 47] is then used to generate the final model based on this alignment. T0280_1 is a Comparative Modeling target. The template used by LOOPP 1I5E_A can be identified by PSI_BLAST with an E value of 2e-028. However, PSI_BLAST fails to generate an accurate alignment between the target and the template. Its alignment has a Q value of 41% and an average shift of 16 residues with respect to the CE structural alignment (Q and shift have the same definition as in the CASP5 test set). SSALN, on the other hand, is able to generate a much better alignment with a Q score of 73% and an average shift of 0.8 residues. T0280 has two domains, with a second domain inserted into the middle of the first domain and this second domain is not present in the template. Figure 8 shows the CE structural alignment between T0280 and 1I5E_A. There is a long gap region in 1I5E_A corresponding to the insertion of the second domain in T0280. SSALN successfully predicts the majority of this gap region and aligns both segments of the first domain (residues 5-52, and 115-179) accurately with the template. PSI_BLAST fails to predict the existence of this gap region. It aligns the region 115-179 correctly, but shifts the region 5-52 by an average of 43 residues. Figure 9A and 9B show the overlap of target T0280 with the MODELLER generated model based on our alignment and the model based on PSI_BLAST alignment respectively. Our predicted model has an rmsd of 1.9Å in domain 1 region, while the PSI_BLAST-based model has an rmsd of 8.1Å in the same region.

Conclusion and Discussion

In template-based modeling of protein structures, the generation of an accurate alignment between a target and its template is a critical step that affects the quality of the final model greatly. Although sequence alignment algorithms can generate accurate alignments when the sequence similarity between the target and the template is high, it still remains an open problem how to reliably produce an accurate alignment when the sequence similarity is low. Here we present a method that performs significantly better than sequence-based methods in all our test sets. We

derive component substitution matrices and position-specific gap penalties based on statistics of structurally aligned protein pairs. The component matrices and gap penalties take into account the amino acid type, secondary structure and solvent accessible area of a position in the scoring. These substitution matrices and gap penalties are then combined linearly with the weights selected as the choice that optimizes the total number of correctly aligned positions in a validation set. In the CASP5 targets-based test set, our method is more than 40% better than sequence-based methods. In an analysis of the contribution from the components, we find that inclusion of the substitution matrix SD (SABLE predicted secondary structure and solvent accessibility of a target position *vs* DSSP assigned secondary structure and solvent accessibility of a template position) plays an important role in the performance improvement. In particular, it helps to lower the average shift from 18 residues to 6 residues. Our method also generates better alignments than PSI_BLAST for the CASP6 test set. In the benchmark set ProSup, our method is better than sequence-based methods, and the threading algorithms that we have tested, SPARKS and the threading method in [42]. In the Fischer's benchmark, our method performs better than CLUSTALW and GenTHREADER, and performs better than FUGUE for the accuracy thresholds > 50%, > 60% and > 70%, but worse than FUGUE for the >80% accuracy threshold.

It has been shown that the evolutionary information present in sequence profiles help to generate better alignments, and profile-profile alignment algorithms have been reported to perform even better than sequence-profile methods such as PSI_BLAST[38; 48; 49]. In this study of the CASP5 test set, PSI_BLAST also performs better than the Smith-Waterman algorithm with Blosum50 matrix. For the templates, we have not only the sequence profile information available, but also the structural profile information based on multiple structural alignments of structural families. In both FUGUE and 3D-PSSM, amino acid types at equivalent structural positions are retrieved to generate a structural profile based on a multiple structural alignment of the template. This structural profile is then aligned against a sequence profile from a

multiple sequence alignment of the target sequence with dynamic programming[19; 20]. SSALN did not directly use the profile information in the alignment algorithm. One direction to improve SSALN is to generate sequence profile of the target, and sequence and structure profiles of the template, and develop position-specific scoring matrices depending on amino acid type, secondary structure and solvent accessibility probability distributions of the two positions being aligned.

Acknowledgements

This research is supported by an NIH grant GM067823. We thank J. Pillardy and T. Galor-Naeh for their help, and Domingues et. al. for making their ProSup benchmark available on line. We thank J. Meller and M. Wagner for providing the program SABLE and PF3.

Table 1. The Alignment Accuracy of the CASP5 test set

Method	Nc	Q(%)	Shift
SW_Blosum50	5315	27.1	23.0
PSI_BLAST	7096	34.7	12.1
SSALN	10416 (10169)	51.3 (50.3)	6.5 (5.9)
Matrix AA	6554 (5922)	33.3 (30.0)	18.8 (20.6)
Matrix AA+AD	7178 (6540)	36.3 (32.7)	18.3 (18.9)

SW_Blosum50: Smith-Waterman algorithm with Blosum50 matrix, and a gap opening penalty of 10 and a gap extension penalty of 1.

SSALN: This study with all three component substitution matrices AA, AD and SD.

Matrix AA: The amino acid type substitution matrix in SSALN.

Matrix AA+AD: The SSALN method without the matrix SD.

The numbers before the parentheses are the results with the position-specific gap penalties developed in this study, and the numbers in the parentheses are the results of the corresponding methods with a constant gap opening penalty of 1.5, and a constant gap extension penalty of 0.15.

Table 2. Alignment accuracy results on the ProSup benchmark

Method	T_c	T_m	T_i	σ_0
SSALN[a]	9256	1115	7245	58.3
Threading[b]				
Global	7692	1291	8914	48.0
Local	7567	1665	7956	47.5
Double gap	7511	1690	7949	47.7
Sequence[b]				
Global	5717	1848	8926	34.0
Local	5719	2390	8148	34.1
Double gap	5699	2392	8144	34.1
FASTA[b]	5340	3003	7452	31.4

a: This study.

b: Results from Domingues et. al[42].

Table 3. Fraction of Correctly Aligned Residues for ProSup benchmark

Method	Sequence[a]	Threading[a]	PSI_BLAST[b]	Stroma[b]	SPARKS[c]	SSALN[d]
σ_0(%)	34.1	48.0	35.6	36.1	57.2(51.4)	58.3

a: Results from Domingues et. al[42].

b: Results from [6].

c: Results from Zhou et. al.[6]. The number in parentheses is the result of SPARKS without the single-body structure-derived score.

d. This study.

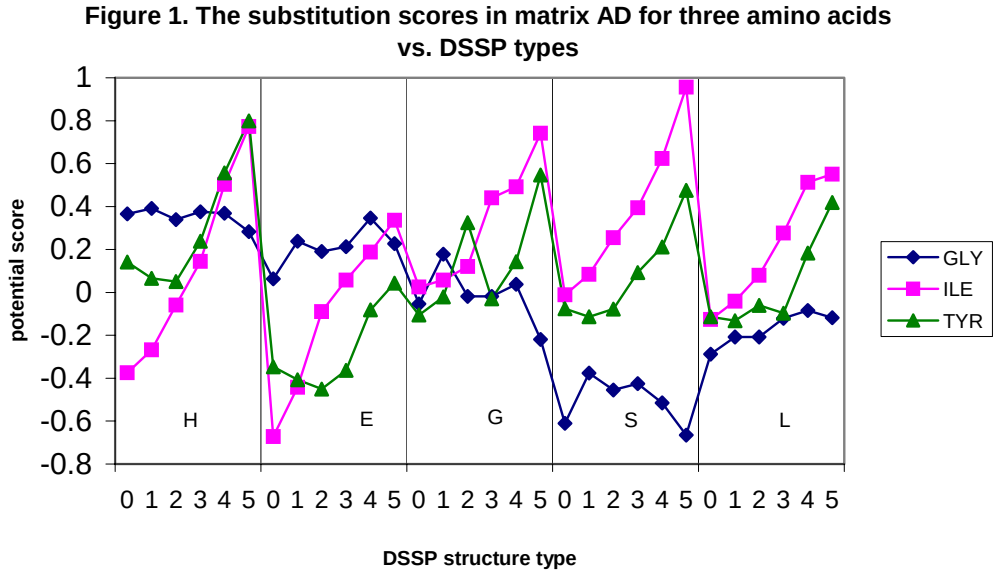


Figure 1. The score of an amino acid as a function of exposed surface (DSSP type – 0 is the most buried) and secondary structure types (e.g. H – helix, E – extended chain)

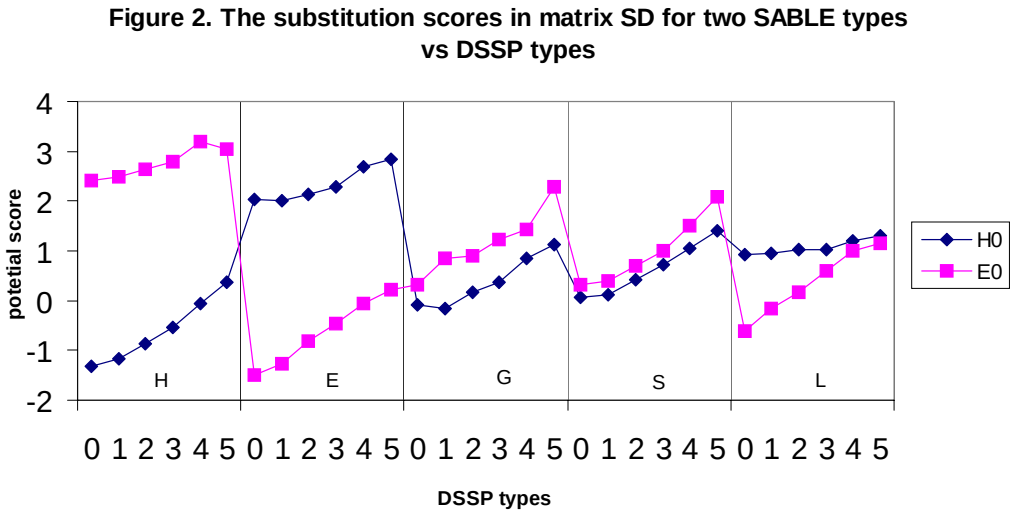


Figure 2. The score of an amino acid as a function of exposed surface (DSSP type – 0 is the most buried) and secondary structure types (e.g. H – helix, E – extended chain)

Figure 3. The DSSP type-dependent gap opening penalties $g_o(D_x)$

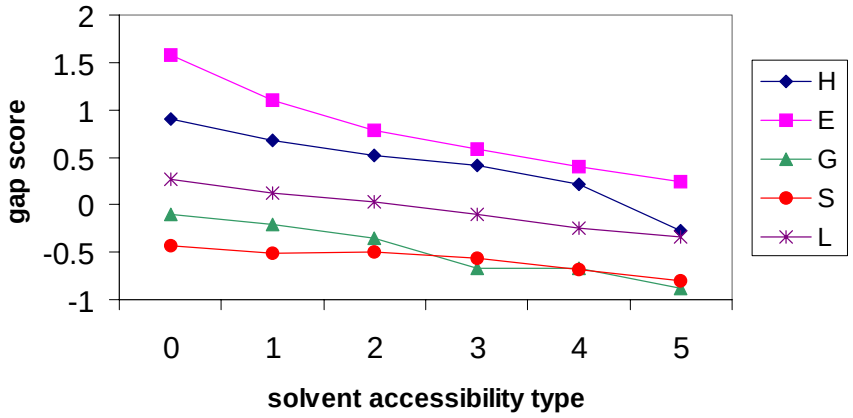


Figure 3. gap penalties as a function of exposed surface area. See text for more details

Figure 4. SABLE type-dependent gap penalty scores $g(SA_n, SA_{n+1})$

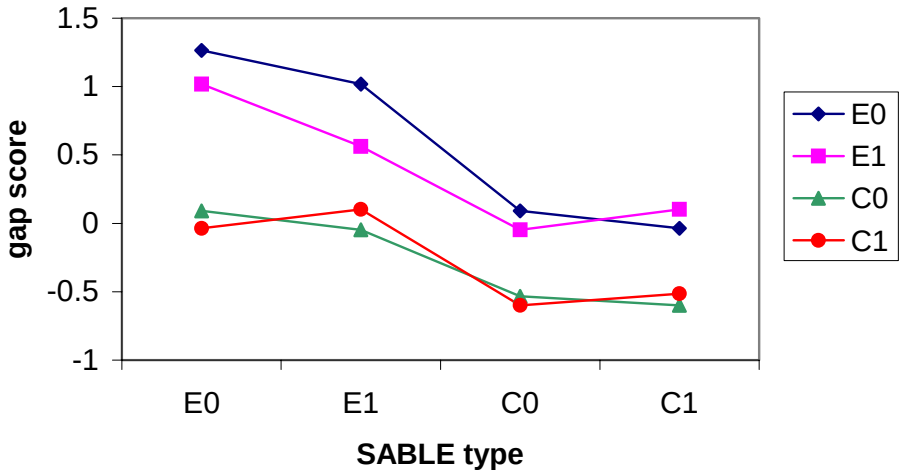


Figure 4. gap penalty as a function of secondary structure types. See text for more details.

Figure 5. The fraction of correctly aligned residues (Q) is plotted against the percent sequence identity between the target and the template for methods SW_BLOSUM50, PSI_BLAST and SSALN (this study). All three methods have a better performance with a higher percent sequence identity. SSALN is notably better than the other two methods in the low sequence similarity region.

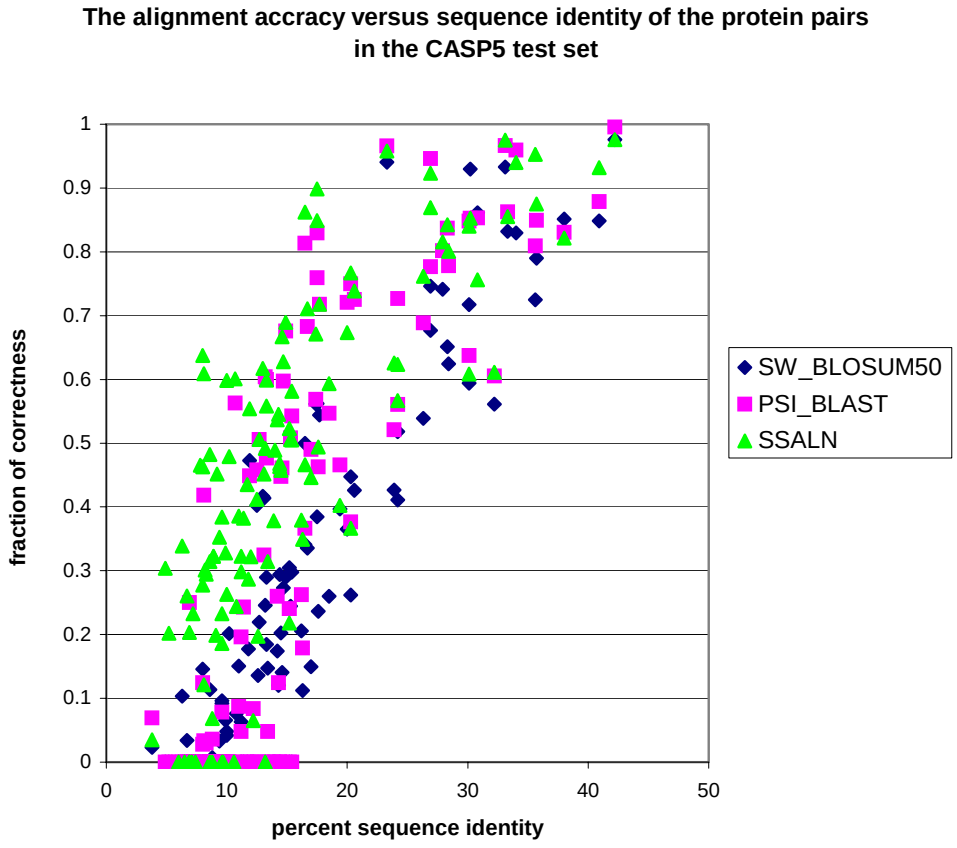
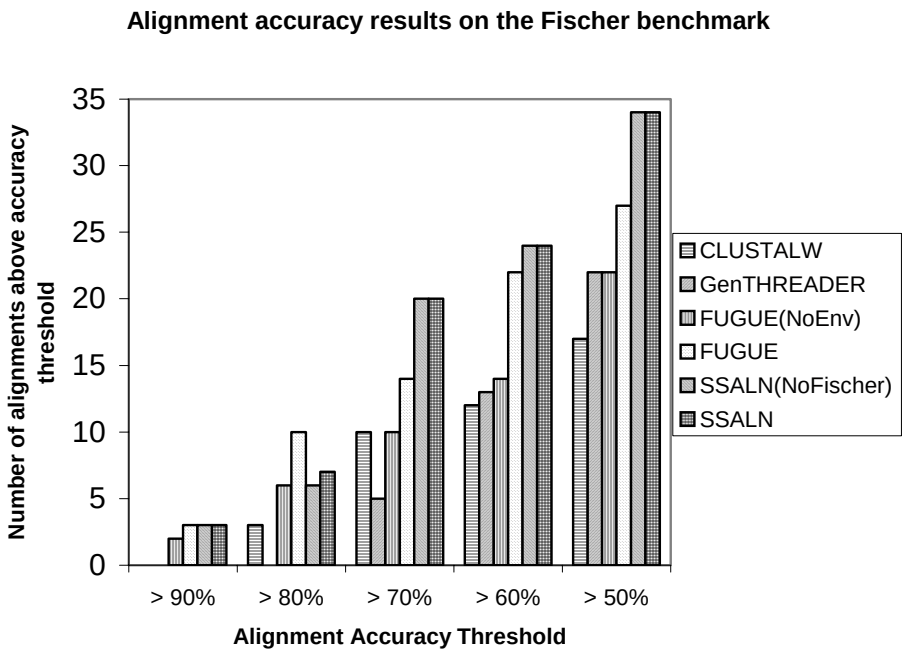


Figure 6. Alignment accuracy comparison between SSALN and three other methods, CLUSTALW, FUGUE, and GenTHREADER on the Fischer's 68-pair benchmark. Number of alignments above five different accuracy thresholds (> 50% ~ > 90 %) is plotted. FUGUE (NoEnv) is the same as FUGUE except that it uses an environment-independent substitution table. SSALN (NoFischer) is the same as SSALN except that proteins present in the Fischer's benchmark are removed from the training set. The results of CLUSTALW and FUGUE are taken from[19] and that of GenTHREADER is taken from[50].



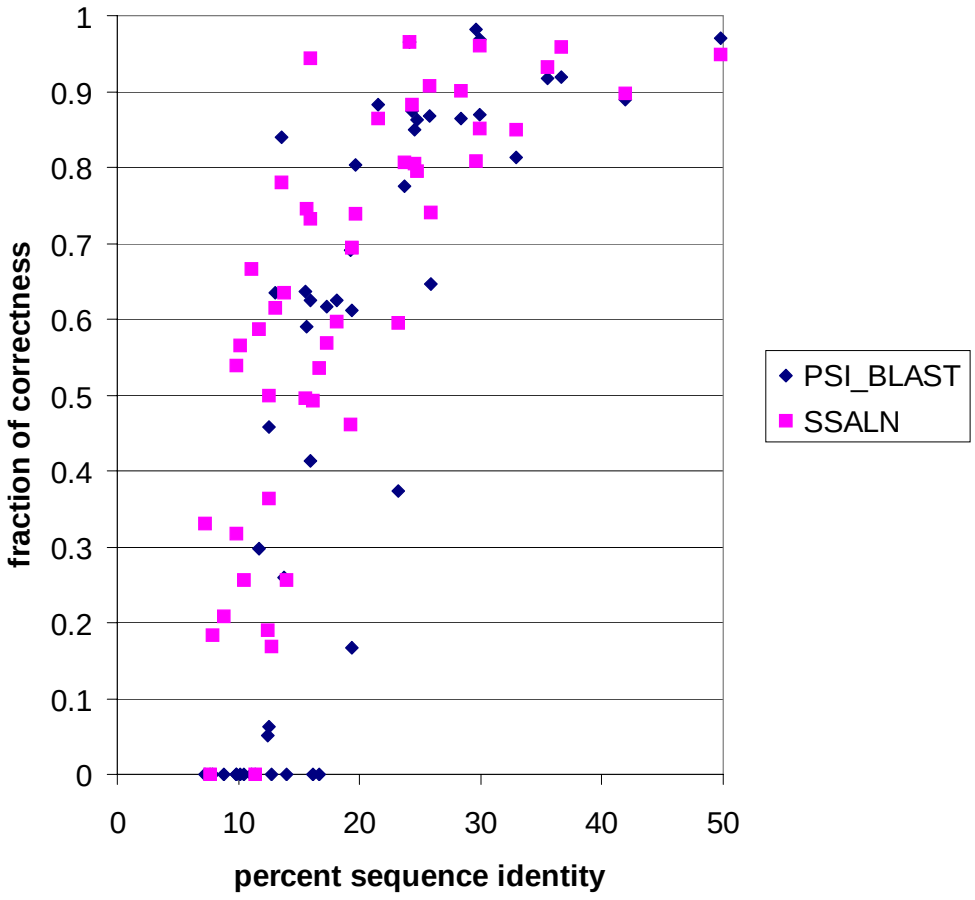


Figure 7A. The alignment accuracy Q_0 versus sequence identity of the protein pairs in the CASP6 test set

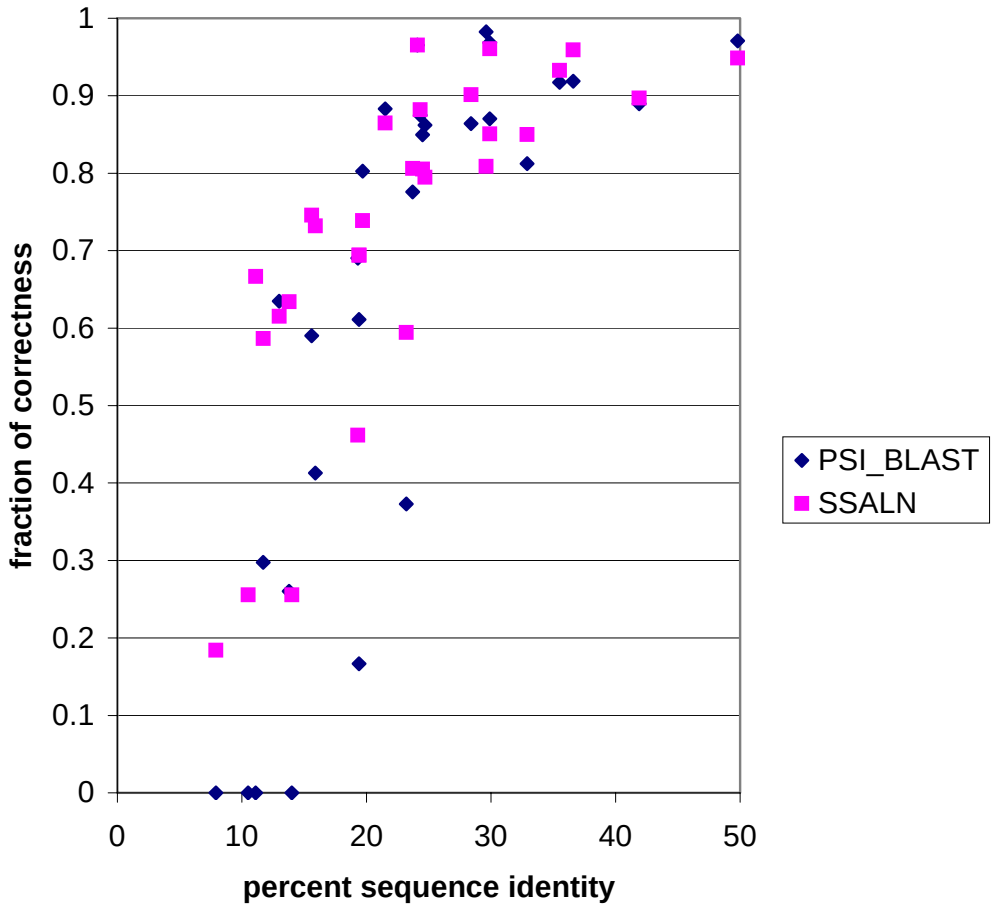


Figure 7B. The alignment accuracy Q_0 of CASP6 targets that LOOPP successfully identified the templates

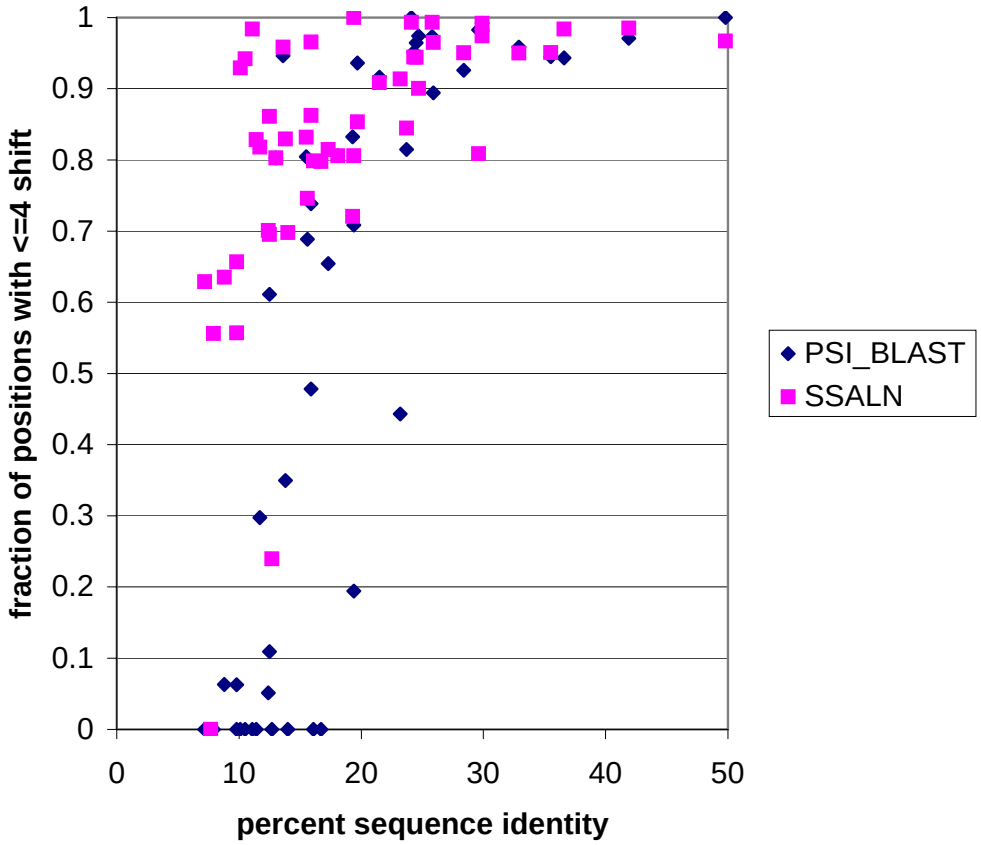


Figure 7C. The alignment accuracy Q_4 versus sequence identity of the protein pairs in the CASP6 test set

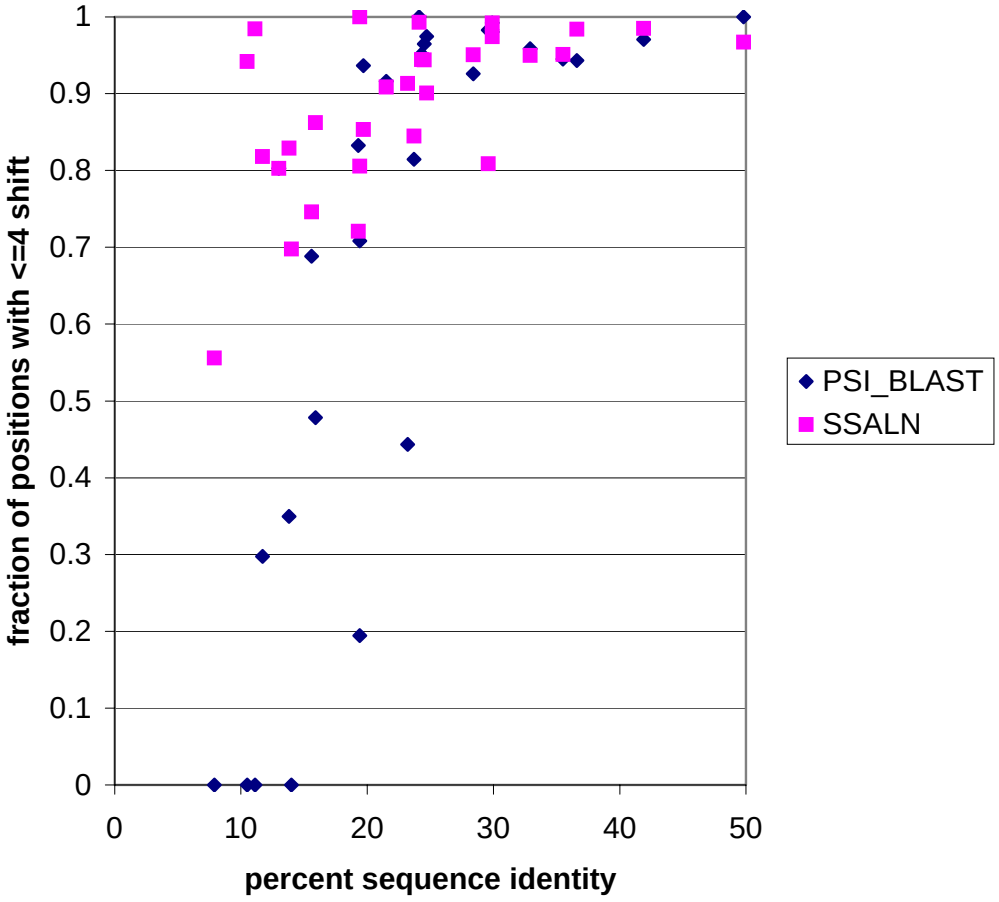


Figure 7D. The alignment accuracy Q_4 of CASP6 targets that LOOPP successfully identified the templates

Figure 8. The CE alignment between the CASP6 target T0280 and its template used in LOOPP, 1I5E_A. The bars between the two sequences indicate the positions aligned in the alignment generated by SSALN. T0280 contains two domains, T0280_1: 5-52, 115-179; T0280_2: 53-103. Domain 1 is present in 1I5E_A, but domain 2 is not, as evidenced by the long gap region in 1I5E_A in the CE alignment.

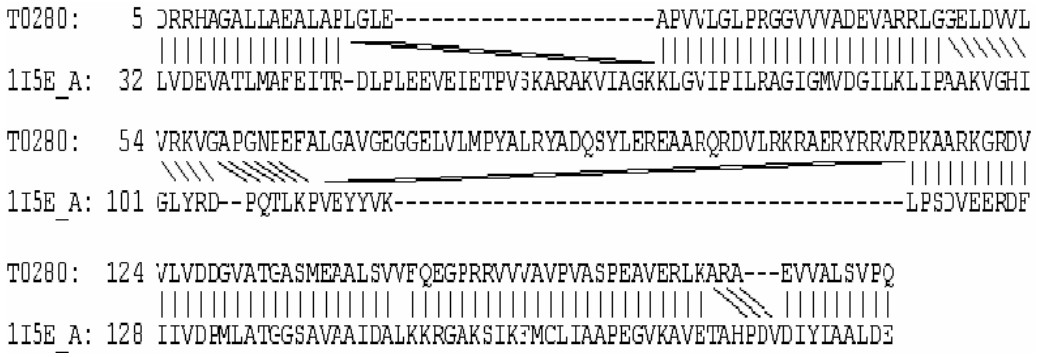
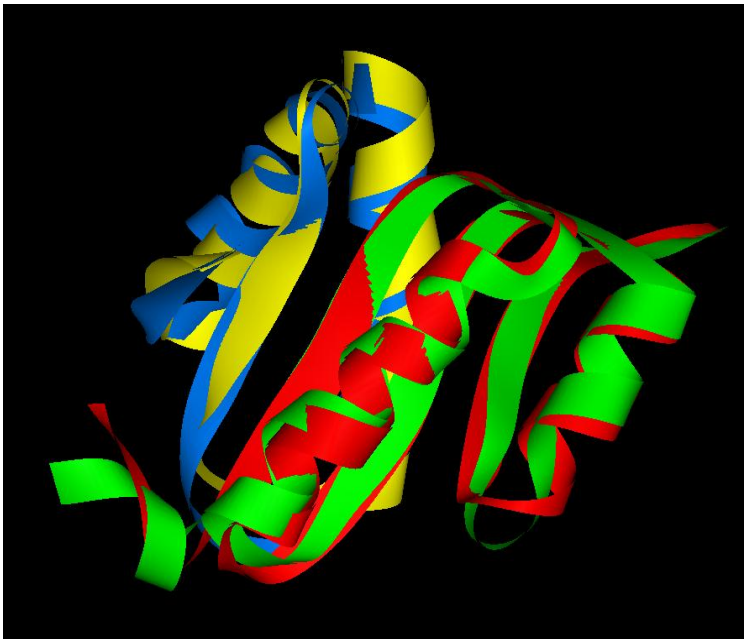
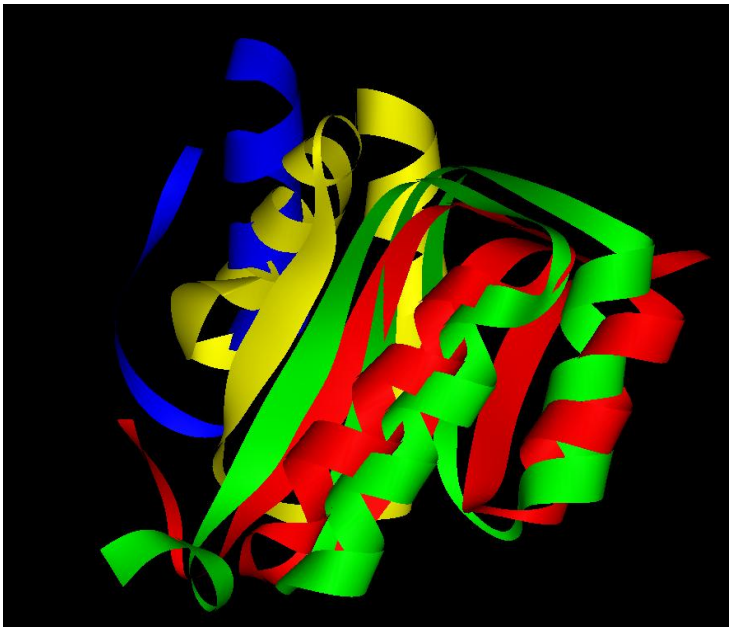


Figure 9. A) The superposition between T0280_1 and the corresponding region in a model generated with MODELLER based on SSALN alignment. The rmsd is 1.9Å in this region. B) The superposition between T0280_1 and the corresponding region in a model generated with MODELLER based on PSI_BLAST alignment. The rmsd is 8.1Å in this region. In both figures, The yellow and red colors represent the regions of residues 5-52 and residues 115-179 in T0280, respectively. The blue and green colors represent the regions of residues 5-52 and residues 115-179 in the MODELLER-generated models, respectively. The PSI_BLAST based model has the region of 5-52 predicted completely wrong due to the erroneous alignment in this region.

A)



B)



References

1. Kinch, L. N., Qi, Y., Hubbard, T. J. & Grishin, N. V. (2003). CASP5 target classification. *Proteins* 53 Suppl 6, 340-51.
2. Venclovas, C. (2003). Comparative modeling in CASP5: progress is evident, but alignment errors remain a significant hindrance. *Proteins* 53 Suppl 6, 380-8.
3. Marti-Renom, M. A., Stuart, A. C., Fiser, A., Sanchez, R., Melo, F. & Sali, A. (2000). Comparative protein structure modeling of genes and genomes. *Annu Rev Biophys Biomol Struct* 29, 291-325.
4. Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Eng* 12, 85-94.
5. Meller, J. & Elber, R. (2001). Linear programming optimization and a double statistical filter for protein threading protocols. *Proteins* 45, 241-61.
6. Zhou, H. & Zhou, Y. (2004). Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins* 55, 1005-13.
7. Bowie, J. U., Luthy, R. & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science* 253, 164-70.
8. Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol* 147, 195-7.
9. Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 48, 443-53.
10. Tobi, D. & Elber, R. (2000). Distance-dependent, pair potential for protein folding: results from linear optimization. *Proteins* 41, 40-6.
11. Lu, H. & Skolnick, J. (2001). A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins* 44, 223-32.
12. Samudrala, R. & Moult, J. (1998). An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *J Mol Biol* 275, 895-916.
13. Lathrop, R. H. (1994). The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng* 7, 1059-68.
14. Bryant, S. H. (1996). Evaluation of threading specificity and accuracy. *Proteins* 26, 172-85.
15. Godzik, A., Kolinski, A. & Skolnick, J. (1992). Topology fingerprint approach to the inverse protein folding problem. *J Mol Biol* 227, 227-38.

16. Karchin, R., Cline, M., Mandel-Gutfreund, Y. & Karplus, K. (2003). Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins* 51, 504-14.
17. McGuffin, L. J. & Jones, D. T. (2003). Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics* 19, 874-81.
18. Rice, D. W. & Eisenberg, D. (1997). A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. *J Mol Biol* 267, 1026-38.
19. Shi, J., Blundell, T. L. & Mizuguchi, K. (2001). FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J Mol Biol* 310, 243-57.
20. Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J Mol Biol* 299, 499-520.
21. Elofsson, A. (2002). A study on protein sequence alignment quality. *Proteins* 46, 330-9.
22. Al-Lazikani, B., Sheinerman, F. B. & Honig, B. (2001). Combining multiple structure and sequence alignments to improve sequence detection and alignment: application to the SH2 domains of Janus kinases. *Proc Natl Acad Sci U S A* 98, 14796-801.
23. Zhang, Z., Lindstam, M., Unge, J., Peterson, C. & Lu, G. (2003). Potential for dramatic improvement in sequence alignment against structures of remote homologous proteins by extracting structural information from multiple structure alignment. *J Mol Biol* 332, 127-42.
24. Russell, R. B., Saqi, M. A., Bates, P. A., Sayle, R. A. & Sternberg, M. J. (1998). Recognition of analogous and homologous protein folds--assessment of prediction success and associated alignment accuracy using empirical substitution matrices. *Protein Eng* 11, 1-9.
25. Fischer, D. & Eisenberg, D. (1996). Protein fold recognition using sequence-derived predictions. *Protein Sci* 5, 947-55.
26. de la Cruz, X. & Thornton, J. M. (1999). Factors limiting the performance of prediction-based fold recognition methods. *Protein Sci* 8, 750-9.
27. Rost, B., Schneider, R. & Sander, C. (1997). Protein fold recognition by prediction-based threading. *J Mol Biol* 270, 471-80.
28. Panchenko, A. R., Marchler-Bauer, A. & Bryant, S. H. (2000). Combination of threading potentials and sequence profiles improves fold recognition. *J Mol Biol* 296, 1319-31.
29. Henikoff, S. & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A* 89, 10915-9.
30. Teodorescu, O., Galor, T., Pillardy, J. & Elber, R. (2004). Enriching the sequence substitution matrix by structural information. *Proteins* 54, 41-8.
31. Sanchez, R. & Sali, A. (1997). Evaluation of comparative protein structure modeling by MODELLER-3. *Proteins* Suppl 1, 50-8.

32. Kabsch, W. & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577-637.
33. Rost, B. & Sander, C. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins* 19, 55-72.
34. Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 292, 195-202.
35. Rost, B. & Sander, C. (1994). Conservation and prediction of solvent accessibility in protein families. *Proteins* 20, 216-26.
36. Pollastri, G., Baldi, P., Fariselli, P. & Casadio, R. (2002). Prediction of coordination number and relative solvent accessibility in proteins. *Proteins* 47, 142-53.
37. Adamczak, R., Porollo, A. & Meller, J. (2004). Accurate prediction of solvent accessibility using neural networks-based regression. *Proteins* 56, 753-67.
38. Yona, G. & Levitt, M. (2002). Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J Mol Biol* 315, 1257-75.
39. Shindyalov, I. N. & Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* 11, 739-47.
40. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25, 3389-402.
41. Johnson, M. S. & Overington, J. P. (1993). A structural basis for sequence comparisons. An evaluation of scoring methodologies. *J Mol Biol* 233, 716-38.
42. Domingues, F. S., Lackner, P., Andreeva, A. & Sippl, M. J. (2000). Structure-based evaluation of sequence comparison and fold recognition alignment accuracy. *J Mol Biol* 297, 1003-13.
43. Feng, Z. K. & Sippl, M. J. (1996). Optimum superimposition of protein structures: ambiguities and implications. *Fold Des* 1, 123-32.
44. Qian, B. & Goldstein, R. A. (2002). Optimization of a new score function for the generation of accurate alignments. *Proteins* 48, 605-10.
45. Fischer, D., Elofsson, A., Rice, D. & Eisenberg, D. (1996). Assessing the performance of fold recognition methods by means of a comprehensive benchmark. *Pac Symp Biocomput*, 300-18.
46. Sali, A. & Blundell, T. L. (1990). Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *J Mol Biol* 212, 403-28.
47. Sali, A. & Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol* 234, 779-815.
48. Jaroszewski, L., Rychlewski, L. & Godzik, A. (2000). Improving the quality of twilight-zone alignments. *Protein Sci* 9, 1487-96.

49. Ohlson, T., Wallner, B. & Elofsson, A. (2004). Profile-profile methods provide improved fold-recognition: a study of different profile-profile alignment methods. *Proteins* 57, 188-97.
50. Jones, D. T. (1999). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol* 287, 797-815.