ITERATIVE NON-SEQUENTIAL PROTEIN STRUCTURAL ALIGNMENT

Saeed Salem and Mohammed J. Zaki*

Computer Science Department, Rensselaer Polytechnic Institute,
110 8th st. Troy, NY 12180, USA
*Email: {salems, *zaki}@cs.rpi.edu*

Structural similarity between proteins gives us insights on the evolutionary relationship between proteins which have low sequence similarity. In this paper, we present a novel approach called STSA for non-sequential pair-wise structural alignment. Starting from an initial alignment, our approach iterates over a two-step process, a superposition step and an alignment step, until convergence. Given two superposed structures, we propose a novel greedy algorithm to construct both sequential and non-sequential alignments. The quality of STSA alignments is evident in the high agreement it has with the reference alignments in the challenging-to-align RPIC set. Moreover, on a dataset of 4410 protein pairs selected from the CATH database, STSA has a high sensitivity and high specificity values and is competitive with state-of-the-art alignment methods and gives longer alignments with lower *rmsd*. The STSA software along with the data sets will be made available on line at `http://www.cs.rpi.edu/~zaki/software/STSA`.
Keywords: non-sequential alignment, CATH, PDB, RIPC set.

1. INTRODUCTION

Over the past years, the number of known protein structures has been increasing at a relatively fast pace, thanks to advancement in MR spectroscopy and X-ray crystallography. Recently (as of Oct 2007) the number of protein structures in the Protein Data Bank(PDB) [1] has reached 46377. Despite having the structural information about so many proteins, the function of a lot of these proteins is still unknown. Structural similarity highlights the functional relationship between proteins. Moreover, structural similarity between proteins allows us to study evolutionary relationship between remotely homologous proteins (with sequence similarity in the twilight-zone), thus allowing us to look farther in evolutionary time [2]. The goal of protein structural alignment is to find maximal substructures of proteins A and B, such that the similarity score is maximized. The two most commonly used similarity measures are: The coordinate distance-based root mean squared deviation (*rmsd*), which measures the spatial euclidean distance between aligned residues; and the distance matrix based measure that computes the similarity based on intra-molecular distances representing protein structures.

The complexity of protein structural alignment depends on how the similarity is assessed. Kolodny and Linial [3] showed that the problem is NP-hard if the similarity score is distance matrix based. Moreover, they presented an approximate polynomial time solution by discrediting the the rigid-body transformation space. In a more recent work, Xu et al. [4] proposed an approximate polynomial time solution, when the contact map based similarity score is used, using similar democratization techniques. Despite the polynomial time approximate algorithms and as the authors themselves noted, these methods are still too slow to be used in search tools.

There is no current algorithm that guarantees an optimal answer for the pair-wise structural alignment problem. Over the years, a number of heuristic approaches have been proposed, which can mainly be classified into two main categories.

1.1. Dynamic Programming Approach

Dynamic Programming (DP) is a general paradigm to solve problems that exhibit the optimal substructure property [5]. DP-based methods [6, 7, 8, 9, 10] construct a scoring matrix S, where each entry S_{OJ} corresponds to the score of matching the i-Th residue in protein A and the j-Th residue in protein B. Given a scoring scheme between residues in the two proteins, dynamic programming finds the global alignment that maximizes the score. Once the best equivalence is found, a superposition step is performed to find the transformation that minimizes the *rmsd* between the corresponding residues. In STRUCTAL [7], the structures are first superimposed onto each other using initial seeds (random or sequence-based). The similarity score S_{OJ} of match-

*Corresponding author.

ing the residues is a function of the spatial displacement between the residue pairs in the superimposed structures. DP is applied on the scoring matrix to get an alignment. The alignment obtained is an initial seed and the process of superposition and alignment is repeated till convergence.

Other methods employed local geometrical features to calculate the similarity score. CTSS [11] used a smooth spline with minimum curvature to define a feature vector of the protein backbone which is used to calculate the similarity score. Tyagi et al. [10] proposed a DP-based method where the similarity is the substitution value obtained from a substitution matrix for a set of 16 structural symbols. DP-based methods suffer from two main limitations: first, the alignment is sequential and thus non-topological similarity cannot be detected, and second, it is difficult to design a scoring function that is globally optimal [3].

1.2. Clustering Approach

Clustering-based methods [12, 13, 14, 15, 16, 17] seek to assemble the alignment out of smaller compatible (similar) element pairs such that the score of the alignment is as high as possible [18]. Two compatible element pairs are consistent (can be assembled together) if the substructures obtained by elements of the pairs are similar. The clustering problem is NP-hard [19], thus several heuristics have been proposed. The approaches differ in how the set of compatible element pairs is constructed and how the consistency is measured.

In [20], initial compatible triplets are found using geometric hashing. Two compatible triplets are consistent if they have similar transformations, where the transformation is defined such that it can transform one triplet onto the other with minimum distance. DALI [12] finds gapless fragment compatible pairs, which are similar hexapeptide fragments. It then uses a Monte Carlo procedure to combine consistent fragments into a larger set of pairs. The optimization starts from different seeds and the best alignment is reported. Compatible elements in SARF2 [13] are similar secondary structure elements (SSEs) which are obtained by sliding a typical α-helix or β-strand over the C_α trace of the protein. The set of the compatible pairs of the SSEs are filtered based on some distance and angle constraints; the final alignment is obtained by finding the largest set of mutually consistent fragment pairs. In an effort to reduce the search space in clustering methods, CE [14] starts with an initial fragment pair and the alignment is extended by the best fragment that satisfies a similarity criteria. In FATCAT [17], DP is used to chain the fragment pairs.

1.3. Our Contributions

We present STSA[a], an efficient non-sequential pairwise structural alignment algorithm. STSA is an iterative algorithm similar in spirit to the iterative Dynamic Programming(DP)-based methods, yet it employs a different technique in constructing the alignment. Specifically, we propose a greedy chaining approach to construct the alignment for a pair of superposed structures. One limitation of DP-based methods is that they only generate sequential alignments. Another limitation is the fact that we do not yet know how to design a scoring function that is globally optimal [3]. Our approach addresses these challenges by looking directly at the superposed structures and assembles the alignment from small closely superposed fragments. Unlike DP, this greedy approach allows for non-topological (non-sequential) similarity to be extracted.

We employ PSIST [21] to generate a list of similar substructures which serve as the initial alignment seeds. Our approach is decoupled such that we can use initial alignment seeds from other methods. In fact, we use SCALI seeds [16] for the RIPC results.

To assess the quality of the STSA alignment, we tested it on the recently published hard-to-align RIPC set [22]. STSA alignments have higher agreement (accuracy) with the reference alignment than state-of-the-art methods: CE, DALI, FATCAT, MATRAS, CA, SHEBA, and SARF. Moreover, we compiled a dataset of 4410 protein pairs from the CATH classification [23]. We measured the overall sensitivity and specificity of STSA to determine if two proteins have the same classification. Results from the CATH dataset indicate that STSA achieves high sensitivities at high specificity levels and is competitive to well established structure comparison methods like DALI, STRUCTAL, and FAST, as judged by the geometric match measure SAS_k [6].

[a]an acronym of **ST**ructural pair-wi**S**e **A**lignment

2. STSA ALIGNMENT

Our approach is based on finding an alignment based on initial seeds. We first discuss how to get the initial seeds and then explain our greedy chaining algorithm.

2.1. Alignment Seeding

The initial alignment seeds are similar substructures between protein A and protein B. An initial seed is an equivalence between a set of pairs. We obtain the seeds from our previous work PSIST [21]. PSIST converts each protein structure into a **S**tructure-**F**eature (**SF**) sequence and then uses suffix tree indexing to find the set of maximal matching segments (initial seeds)

Another source of seeds we use is the SCALI seeds [16]. The SCALI seeds are gapless local sequence-structure alignments obtained using HMM-STR [24], which is an HMM built on top of a library of local motifs.

An initial seed $s = F_i^A F_j^B(l)$ indicates that the fragment of protein A that starts at residue i matches the fragment from protein B that starts at residue j and both the fragments has equal length of l.

2.2. Iterative Superposition-Alignment Approach

Each alignment seed $(F_i^A F_j^B(l))$ is treated as an initial equivalence, E_0, between a set of residues from protein A and a set of residues from protein B. The correspondence between the residues in the equivalence is linear, i.e. $E = \{(a_i, b_j), \cdots, (a_{i+l-1}, b_{j+l-1})\}$. Given an equivalence E, we construct an alignment of the two structures as follows.

2.2.1. *Finding Optimal Transformation*

We first find a transformation matrix T_{opt} that optimally superposes the set of pairs of residues in the equivalence E such that the *rmsd* between the superposed substructures of A and B is minimized:

$$T_{opt} = arg_{min}(T) \;\; RMSD_T(E) \ ,$$

where $RMSD_T(E) = \frac{1}{|E|}\sum_{(i,j)\in E} d(T[a_i], b_j)$. We find the optimal transformation T_{opt} using the Singular Value Decomposition [25, 26].

2.2.2. *Constructing Scoring Matrix*

We next apply the optimal transformation T_{opt} obtained in the previous step to protein A to obtain A^*. We then construct a $n \times m$ binary scoring matrix S, where n and m denote the number of residues in proteins A and B, respectively and $S_{ij} = score(dist(a_i^*, b_j))$; the score is 1 if the distance between corresponding elements, a_i^* and b_j is less than a threshold δ, and 0 otherwise.

2.2.3. *Finding an Alignment*

An alignment is a set of pair of residues $\{(a_i, b_j)\}$, a_i in A, and b_j in B. Based on the scoring matrix S we find the maximum correspondence by finding the maximum cardinality matching in the bipartite graph $G(U, V, E)$ where U is the set of residues in protein A, V is the set of residues in proteins B, and there is an edge $(a_i, b_j) \in E$ if $S_{ij} = 1$. However, the problem with the maximum matching approach is that it may yield several short, disjoint and even arbitrary matchings that may not be biologically very meaningful. Our goal is to find an alignment composed of a set of segments such that each segment has at least r residue pairs.

A run R_i is a set of consecutive diagonal 1's in the scoring matrix S which constitutes an equivalence, between a substructure in A and another in B, that can be aligned with a small *rmsd*. Specifically, a run R is a triplet (a_i, b_j, l), where a_i is the starting residue for the run in A (similarly b_j for B), and the the length of the run is l. The correspondence between residues in the run is as follows: $\{(a_i, b_j), \cdots, (a_{i+l-1}, b_{j+l-1})\}$.

The matrix S has a set of runs $R = \{R_1, R_2, \cdots, R_k\}$ such that $|R_i| \geq r$, where r is the minimum threshold length for a run. We are interested in finding a subset of runs $C \subseteq R$ such that all the runs in C are mutually non-overlapping and the length of the runs in C, $L(C) = \sum_{i\in C}|R_i|$ is as large as possible. The problem of finding the subset of runs with the largest length is essentially the same as finding the maximum weighted clique in a graph $G = (V, E)$ where V is the set of runs, with the weight for vertex i given as $w_i = |R_i|$, and there is an edge $(i, j) \in E$ if the runs R_i and R_j do not overlap. The problem of finding the maximum weighted clique is NP-hard [19], therefore we use greedy algorithms to find an approximate solution. Note that it is also possible to use a dynamic programming approach to align the proteins based on the scoring matrix S, however, this would yield only a sequential alignment. Since we are interested in non-sequential alignments, we adopt the greedy weighted clique fining approach.

The simplest greedy algorithm chooses the longest run $R_i \in R$ to be included in C, and then removes from R all the runs R_j that overlap with R_i. It then chooses the longest remaining run in R, and iterates this process until R is empty. We also implemented an enhanced greedy algorithm that differs in how it chooses the run to include in C. It chooses the run $R_i \in R$ that has the highest weight w where $w(R_i)$ is the length of R_i plus the lengths of all the remaining non-overlapping runs. In other words, this approach not only favors the longest run, but also favors those runs that do not preclude many other (long) runs.

Through our experiments, we found that the simple greedy algorithm gives similar alignments in terms of the length and *rmsd* as the enhanced one. Moreover, it is faster since we do not have to calculate the weights every time we choose a run to include to C. Therefore, we adopt the first heuristic as our basic approach. Note that it is also possible to use other recently proposed segment chaining algorithms [27]. The subset of runs in C makes up a new equivalence E_1 between residues in proteins A and B. The length of the alignment is the length of the equivalence $|E_1| = \sum_{i \in C} |R_i|$ and the *rmsd* of the alignment is the *rmsd* of the optimal superposition of the residue pairs in E_1.

2.2.4. *Refining the Alignment*

To further improve the structural alignment we treat the newly found equivalence E_1 as an initial alignment and repeat the previous steps all over again. The algorithm alternates between the superposition step and the alignment step until convergence (score does not improve) or until a maximum number of iterations has been reached. Figure 1 shows the pseudo-code for our iterative superposition-alignment structural alignment algorithm. The method accepts the set of maximal matching segments $M = \{F_i^A F_j^B(l)\}$ as initial seeds. It also uses three threshold values: δ for creating the scoring matrix, r for the minimum run length in S, and L for the maximum *rmsd* allowed for an equivalence. For every initial seed we find the optimal transformation (lines 4-5), create a scoring matrix (line 6), and derive a new alignment E_1 via chaining (line 7). If the *rmsd* of the alignment is above the threshold L we move on to the next seed, or else we repeat the steps (lines 3-10) until the score no longer improves or we exceed the maximum number of iterations. The best alignment found for each seed is stored in the set of potential alignments $\mathcal{E}$ (line 11). Once all seeds are processed, we output the best alignment found (line 13). We use the SAS_k [6] geometric match measure (explained in the next section) to score the alignments. We noticed that typically three iterations were enough for the convergence of the algorithm.

```
$M = \{F_i^A F_j^B(l)\}$, set of seed alignments
$L$, the rmsd threshold
$r$, the min threshold for the length of a run in $S$
$\delta$, the max distance threshold for $S$

SEED-BASED ALIGNMENT $(M, L, r, \delta)$:
1.  for every $F_i^A F_j^B(l) \in M$
2.      $E$ is the equivalence based on $F_i^A F_j^B(l)$
3.      repeat
4.          $T_{opt} = RMSD_{opt}(E)$
5.          $A^* = T_{opt} A$
6.          $S_{ij} = 1$ if $d(a_i^*, b_j) < \delta$, 0 otherwise
7.          $E_1$= chain-segments($S$, $r$)
8.          if $RMSD_{opt}(E_1) \geq L$ go to step 2
9.          $E \longleftarrow E_1$
10.     until score does not improve
11.     add $E$ to the set of alignments $\mathcal{E}$
12. end for
13. Output best alignment from $\mathcal{E}$
```

Fig. 1. The STSA Algorithm

2.3. Scoring the alignments

We assess the significance of STSA alignments by using the geometric match measure, SAS_k, introduced in [6], defined as follows:

$$SAS_k = rmsd(100/N_{mat})^k$$

where *rmsd* is the coordinate root mean square deviation, N_{mat} is the length of the alignment, and k is the degree to which the score favors longer alignments at the expense of *rmsd* values. In our implementation, we use $k = 1$, $k = 2$ and $k = 3$ to score the alignments, to study the effect of the scoring function. For each of the three scoring schemes SAS_1, SAS_2 and SAS_3, a lower score indicates a better alignment, since we desire lower *rmsd* and longer alignment lengths. Kolodny et al. [28] recently contended that scoring alignment methods by geometric measures yields better specificity and sensitivity; we observe consistent behavior in our results.

2.4. Initial Seeds Pruning

Since the quality of the alignment depends on the initial alignment (seed), we start with different initial seeds in an attempt to reach a global optimum alignment. This, however, results in a slow algorithm since we could potentially have a large number of initial seeds. Let the size of protein A be n and of B be m, respectively and $n \leq m$. The number of maximal matching segments can be as large as nm/l_{min}, where l_{min} is the length threshold. Most of these seeds do not constitute good initial seeds as judged by their final global alignments. In order to circumvent this problem, we only select heuristically the most promising seeds based on two heuristics: first, the length of the seed; second, the DALI rigid similarity score [12]. In the results section, we study the effect of these pruning heuristics on the quality of the alignments and the improvement in the running time that we gain.

2.5. Computational Complexity

The worst case complexity of finding the maximal matching segments using PSIST is $O(nm)$, where m and n denote the lengths of proteins A and B [21]. Assuming $m \leq n$, the complexity of constructing the full set of runs R is $O(nm)$, since we have to visit every entry of the scoring matrix. Since we use a threshold of $\delta = 5\mathring{A}$ to set $S_{ij} = 1$ in the scoring matrix, each residue, due to distance geometry, in A can be close to only a few residues in B (after superposition). Therefore, there are $O(n)$ 1's in the matrix S. And thus, we have d$O(n)$ diagonal runs, and sorting these runs takes $O(n \log n)$ time. In the greedy chaining approach, for every run we choose, we have to eliminate other overlapping runs, which can be done in $O(n)$ time per check, for a total time of $O(n^2)$. Over all the steps the complexity of our approach is therefore $O(n^2)$.

3. RESULTS

To assess the quality of STSA alignments compared to other structural alignment methods, we tested our method on the hard-to-align RIPC set [22]. Moreover, we evaluated the overall sensitivity and specificity of STSA compared to other alignment methods over 4410 alignment pairs using the CATH [23] classification as a gold standard.

The criteria on which we selected the other algorithms to compare with were: the availability of the program so that we could run it in-house, and the running time of the algorithm. We compared our approach against DALI [12], STRUCTAL [6], SARF2 [13], and FAST [15]. For the RIPC dataset, we used the published results for CE [14], FATCAT [17], CA [b], MATRAS [c], LGA [29], and SHEBA [30].

All the experiments were run on a 1.66 GHz Intel Core Duo machine with 1 GB of main memory running Ubuntu Linux. The default parameters for STSA were $r = 3$, $\delta = 5.5\mathring{A}$ and using top 100 initial seeds (see Section 3.3 for more details).

3.1. RIPC set

The RIPC set contains 40 structurally related protein pairs which are problematic to align. Reference alignments for 23 (out of the 40) structure pairs have been derived based on sequence and function conservation. We measure the agreement of our alignments with the reference alignments provided in the RIPC set. As suggested in [22], we compute the percentage of the residues aligned identically to the reference alignment(I_s) relative to the reference alignment's length (L_{ref}).

As shown in Figure 2, while all the methods have mean agreements equal 60 percent or lower, the mean agreement of STSA alignments is 71%. As for the median, all the methods except FATCAT (63%) have median agreements less than 60%, while STSA alignments have a median agreement of 67% .

Some the alignments in the RIPC set are sequential. In these cases, most of the sequential alignment methods return a high agreement with the reference alignment. Thus, in few cases the sequential alignment of STSA gives a higher agreement than the non-sequential alignment. If we were to take the STSA alignment that gives a higher agreement with the reference alignment, then STSA alignments would have a mean and median agreement of 77% and 83% , respectively (STSABest in Figure 2).

As Mayr et al. [22] noted, there are seven chal-

[b] http://bioinfo3d.cs.tau.ac.il/c_alpha_match/
[c] http://biunit.naist.jp/matras/

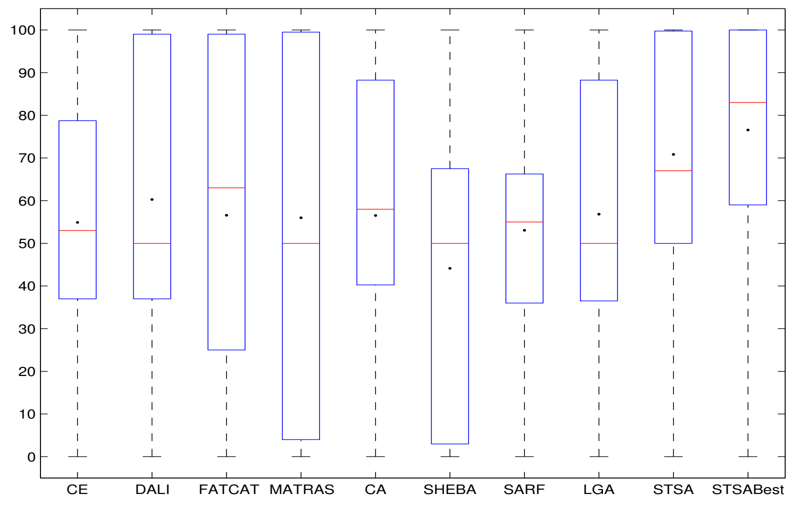


Fig. 2. Comparison of the alignments of 8 methods with the reference alignments from the RIPC set. Box-and-whisker plots for the distribution of agreements of the alignments produced by different methods as compared to the true reference alignments. The dark dot indicates the mean, the horizontal line inside the boxes indicates the median, and the box indicates the range between the lower and the upper quartiles. Results for all the other methods (except SARF) are taken from [22].

lenging protein pairs which reveal how repetition, extensive indels, and circular permutation result in low agreement with the reference alignments. We found two protein pairs particularly problematic to align for all the sequential methods and sometimes the non-sequential ones, except STSA. First, for alignment of L-2-Haloacid dehalogenase (SCOP id: d1qq5a_, 245 residues) with CheY protein (d3chy__, 128 residues), all the methods (except SARF returned 33%) returned zero agreement with the reference alignment while STSA returned 100 percent agreement. The second problematic pair was of the alignment of NK-lysin (d1nkl__, 78 residues) with (Pro)phytepsin (d1qdma1, 77 residues) which has a circular permutation. For the second pair, all the methods (except CA returned 41%, and SARF returned 92%) returned zero agreement with the reference alignment while STSA returned 99 percent agreement. In this pair the N-terminal region of domain d1nkl__ has to be aligned with the C-terminal region of domain d1qdma1 to produce an alignment that matches the reference alignment (see Figure 3). By design, sequential alignment methods cannot produce such an alignment, and therefore fail to capture the true alignment. Among the non-sequential methods, the agreement of STSA alignments with the reference alignments are higher than the agreement of either CA and SARF.

As shown in Figure 3, all the last five methods (DALI, MATRAS, SHEBA, FATCAT, and LGA) have their alignment paths along the diagonal and do not agree with with the reference alignment (shown as circles). The CA method reports a non-sequential alignment that partially agrees with the reference alignment but it misses 59% of the reference alignment pairs. Both SARF and STSA alignments have excellent agreement with the reference alignment, 92%, 99%, respectively.

3.2. Measuring Sensitivity and Specificity using CATH

Gerstein and Levitt [31] emphasized the importance of assessing the quality and significance of structural alignment methods using an objective approach. They used the SCOP database [32] as a gold standard to assess the sensitivity of the structural alignment program against a set of 2,107 pairs that have the same SCOP superfamily. In a more recent work, Kolodny et al. [28] presented a comprehensive comparison of six protein structural alignment methods. They used the CATH classification [23] as a gold standard to compare the rate of true and false positives of the methods. Moreover, they showed that the geometric match measures like SAS_k can better assess the quality of the structural alignment methods. We adopt a similar approach to assess the significance of our approach by comparing the

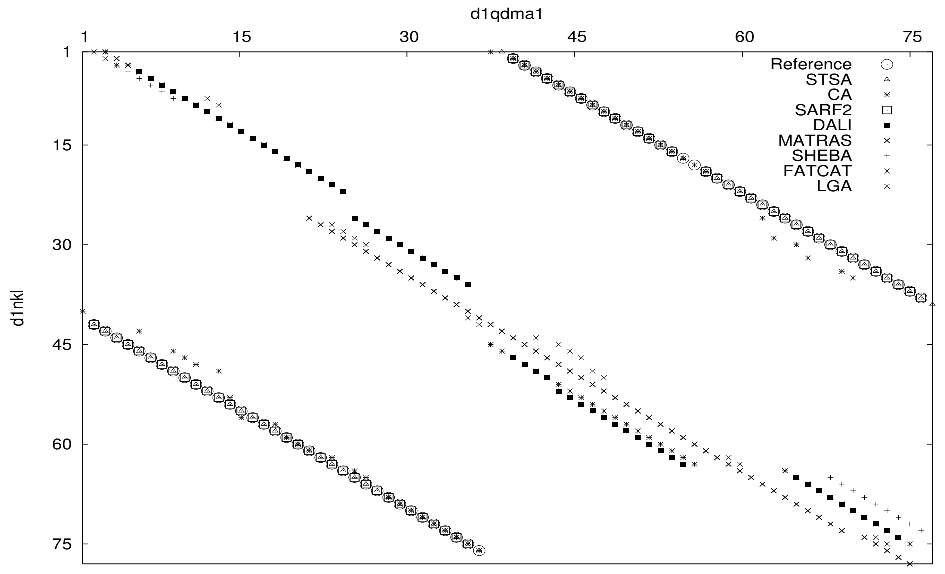


Fig. 3. Comparison of the agreement with the reference alignment of STSA alignment and 6 other alignment methods. Residue positions of d1qdma and d1nkl__ are plotted on the x-axis and y-axis, respectively. Note: The reference alignment pairs are shown in circles. The CA, SARF, and STSA plots overlap with the reference alignment. For this pair, we used the alignment's server of the corresponding method to get the alignment, except for DALI and SHEBA which we ran in-house.

true and false positive rates of STSA alignments to those of other three methods: DALI, STRUCTAL, and FAST. Since the other methods report only sequential alignments, for STSA we also used sequential alignments.

3.2.1. *The CATH Singleton Dataset*

CATH [23] is a hierarchical classification of protein domain clusters. The CATH database clusters structures using automatic and manual methods. The latest version (3.1.0; as for Jan'07) of the CATH database contains more than 93885 domains (63453 chains, from 30028 proteins) classified into 4 Classes, 40 Architectures, 1084 Topologies, and 2091 Homologous Superfamilies. The class level is determined according to the overall secondary structure content. The architecture level describes the shape of the domain structure. The topology (fold family) level groups protein domains depending on both the overall shape and connectivity of the secondary structures. Protein domains from the same homologous superfamily are thought to share a common ancestor and have high sequence identity or structure similarity.

We define protein domains that belong to homologous superfamilies which have only one member as *singletons*. There are 1141 singleton protein domains which belong to 648 different topologies in CATH. Since singleton domains are unique in their homologous subfamily, the structurally closest domains to the singleton domains are the domains in their neighboring H-levels in the same topology. We selected a set of 21 different topologies such that each topology has a singleton subfamily and at least ten other superfamilies. There are only 21 such topologies in CATH, and one domain for each homologous superfamily within a topology is randomly chosen as a representative. So, we have 21 singleton domains and 210 (10×21) domains selected from the different sibling superfamilies. Our final dataset thus has 4410 alignment pairs (21×210). The set of pairs which have the same CATH classification are labeled as positive examples, and as negative examples if they disagree. We have 210 positive pairs and 4200 negative pairs in our dataset.

3.2.2. **Alignment Results**

We ran all the methods on the 4410 structure pairs. The methods report the number of residues in the alignment, the *rmsd*, and the native alignment score: STRUCTAL reports a p-value for the alignment, FAST reports a normalized score, and DALI reports a z-score. For STSA, we score the alignments using the geometric matching score SAS_3. We sort the alignments by the methods' native score and calculate the true positives (TP), i.e., pairs with same

CATH classification, and the false positives (FP), i.e., pairs with a different CATH classification in the top scoring pairs. Moreover, we compare the quality of the alignments of different methods by comparing the average SAS matching score for the true positives.

Figure 4(a) shows the Receiver Operating Characteristic (ROC) curves for all the methods. The ROC graph plots the true positive rate (sensitivity), versus the false positive rate (1-specificity). Recall that the true positive rate is defined as $\frac{TP}{TP+FN}$, and the false positive rate is defined as $\frac{FP}{TN+FP}$, where TP and TN are the number of true positives and negatives, whereas FP and FN are the number of false positives and negatives. All the alignments were sorted by their native score (when applicable), or by the geometric score SAS_3.

Table 1. Comparison of the average alignment length.

TP	DALI	STRUCTAL	FAST	STSA
0.2	100.29/3.06	83.60/2.04	82.52/3.18	101.8 /3.1
	3.05/3.03	**2.44**/3.49	3.85/5.66	3.05/**2.94**
0.4	85.40/3.21	71.67/2.09	70.90/3.16	86.43/3.05
	3.76/5.15	**2.92**/5.68	4.46/8.87	3.53/**4.72**
0.6	75.77/3.36	65.97/2.20	63.90/3.22	76.66/3.03
	4.43/7.72	**3.33**/7.66	5.04/12.34	3.95/**6.73**
0.8	69.49/3.56	64.33/2.49	57.60/3.51	68.48/2.95
	5.12/10.61	**3.87**/9.35	6.09/18.37	4.31/**9.19**
1.0	63.03/3.76	62.09 /2.84	51.75/3.55	61.49/2.88
	5.97/15.02	**4.57/11.86**	6.86/25.62	4.68/12.39

The results are reported as follows: for each sensitivity value, the top row shows the average $N_{mat}/rmsd$, and the bottom row shows SAS_3/SAS_1, where the averages are calculated over the true positive alignments. The values in bold show the best SAS_3 and SAS_1 scores.

Having the best ROC curve does not imply the best alignments. [28] showed that the best methods, with respect to the ROC curves, do not necessarily have the best average geometric match score for the true positives pairs. Our results confirm this observation. Figure 4(b) shows the average SAS_3 measure of the true positives as we vary the number of top k scoring pairs. Clearly, STSA has the best average SAS score for the true positives. This can be explained by the fact that we use the SAS measure in our alignment algorithm. STRUCTAL comes second in the quality of the average SAS measure. Even though FAST was able to classify as many true positives as DALI and STSA, it still has the worst average SAS measure, indicating that it produces shorter alignments with higher *rmsd*. These results suggest is that if the goal is to simply discriminate between the classes, a method can score better than another method that produces better alignments in terms of both length and *rmsd*. However, since our goal is to assess the geometric quality of the alignments, we can clearly see that STSA outperforms the other approaches. Figure 4(c) shows the ROC curve of all the methods after sorting the alignments based on the geometric match score, SAS_3; STSA has the best ROC curve.

In fact, if we use different geometric scoring measures like SAS_2 and SAS_1, we find that STSA continues to give good alignments. Figures 5(a) and 5(c) show the average SAS_2 and SAS_1 scores, respectively, versus the true positive rates, and Figures 5(b) and 5(d) show the corresponding ROC curves. We find that for SAS_2, STSA is still the best. For SAS_1, which emphasizes lower *rmsd* more than length, we find that STRUCTAL is the best method, but is followed closely by STSA.

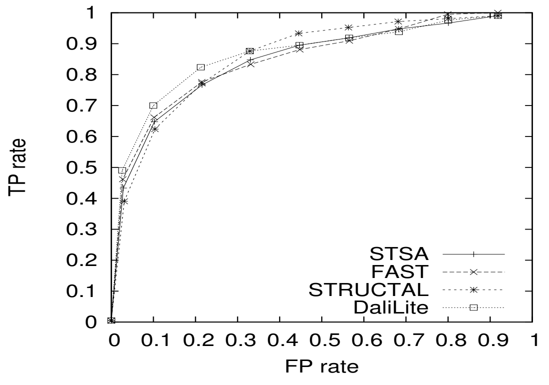
Table 1 summarizes these results in tabular form. It shows the average length and *rmsd* as well as the average SAS_3 and SAS_1 scores, for the true positive alignments for different sensitivities. At all sensitivities, the average STSA alignment length is longer than other methods. This gain in the alignment length comes at little or no cost in terms of the average *rmsd*. Compared to DALI and FAST, STSA is always superior in its alignment quality. Its SAS_3 score is much better (lower) than the other methods. On the other hand, if one prefers shorter, more accurate alignments, then STRUCTAL has the lowest SAS_1 scores, followed by STSA. If fact, by changing the parameters of STSA, we can explicitly bias, it to favor such shorter alignments if those are of more interest.

3.2.3. *Running times*

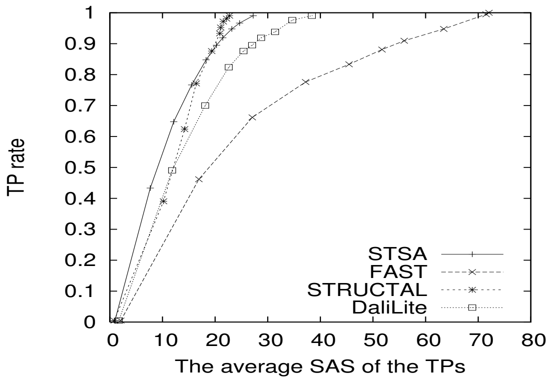
Table 2 shows the total running time for the alignment methods on all the 4410 pairs in the singleton dataset. FAST is extremely fast but its alignments' quality is not so good. STSA is slightly slower than STRUCTAL, but is faster than DALI.

Table 2. Comparison of the running times on the CATH Dataset.

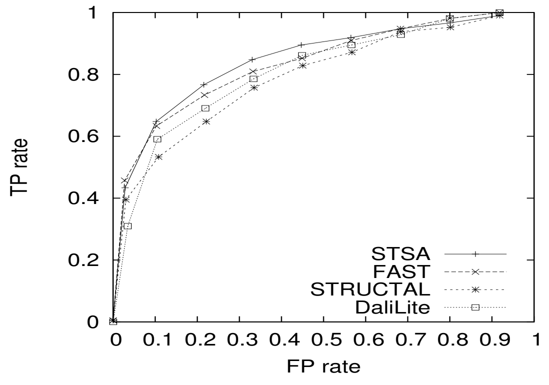
Method	DALI	STRUCTAL	FAST	STSA
Time (s)	4932s	3179s	224s	3893s



(a) ROC: Native Scores



(b) Geometric Score: SAS_3



(c) ROC: SAS_3

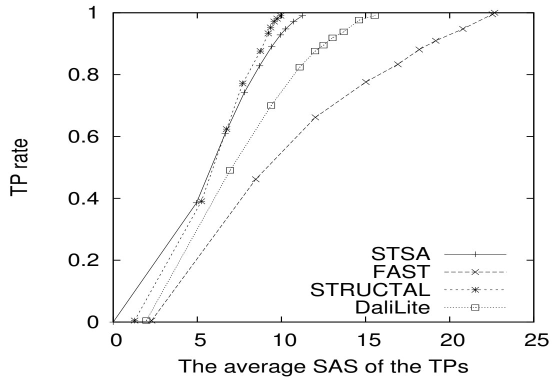
Fig. 4. Receiver Operating Characteristic (ROC) curves for the structural alignment methods measured over the 4410 pairs. (a) The alignments are sorting based on the native score or on the geometric match measure SAS, we tallied the number of true positives and false positives using CATH as a gold standard. (b) The average SAS_3 scores versus the true positive rate. (c) For all the methods, the alignments are sorted using SAS_3 scores and we plot the ROC curve showing the number of true and false positives.

3.3. Analysis of STSA

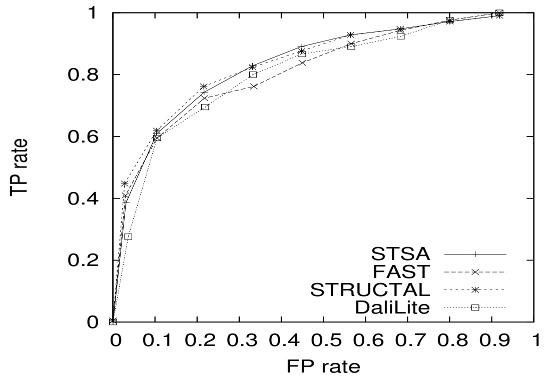
There are some parameters that affect the quality of the resulting alignment in STSA, namely the length of the smallest runs to consider r, and the threshold distance, δ which is used to populate the scoring matrix, and the number of initial seeds. The optimal values for $r = 3$ and $\delta = 5.5$ were found empirically such that they give the best ROC curve on the CATH data set. Here we investigate the effect of seed pruning on the sensitivity of STSA alignments, as well as the quality of the alignments. Figure 6 shows how the average SAS score changes when using different number of initial seeds for the two seed pruning heuristics. The first pruning approach sorts and selects the top k initial seeds based their length (in decreasing order), whereas the second approach uses the DALI rigid similarity scores [12]. Figure 6(a) shows that considering only the top $k = 100$ seeds, the average SAS scores for the true positives are almost as good as using all the seeds. Moreover, as seen in Figure 6(b), using the more sophisticated DALI rigid similarity score to sort the seeds performs the same as using the much simpler and cheaper length-based approach. As for the running time, pruning the seeds and using only the top 100 resulted in a drastic reduction in the running time. As reported in Table 2 STSA took $3893s$ when using the top 100 seeds, whereas it took $9511s$ seconds when using all the seeds.

3.4. Two non-sequential alignments

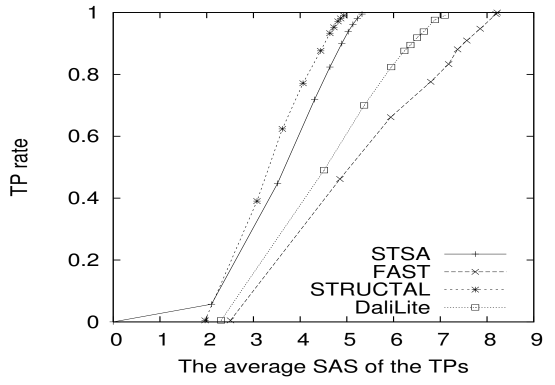
To demonstrate the quality of STSA in finding non-sequential alignments, we present the alignment on a pair of structures reported in SARF2 [13]. Figure 7 shows a non-sequential alignment between Leghemoglobin (2LH3:A) and Cytochrome P450 BM-3



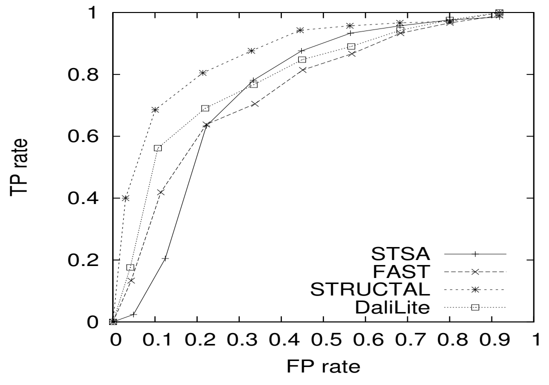
(a) SAS_2



(b) ROC: SAS_2



(c) SAS_1



(d) ROC: SAS_1

Fig. 5. Effect of different geometric matching scores, SAS_k, for $k = 2$ and $k = 1$. (a) The average SAS_2 for the true positive alignments. (b) ROC curve using SAS_2 score. (c) Average SAS_1 for true positives, and (d) ROC using SAS_1 score for sorting alignments.

(2HPD:A). STSA and SARF2 has some common aligned segments, but STSA yielded an alignment of length 114 and $rmsd = 3.37Å$, whereas SARF2 yielded an alignment with length 108 and $rmsd = 3.05Å$. The SAS_3 score of STSA is 2.27, which is better than SARF2's score of 3.84. On this example both SCALI and FAST failed to return an alignment. Also, as expected, this is a hard alignment for sequential alignment methods: STRUCTAL aligned 56 residues with $rmsd = 2.27$, DALI aligned 87 residues with $rmsd = 4.8$, and CE aligned 91 residues with $rmsd = 4.05$.

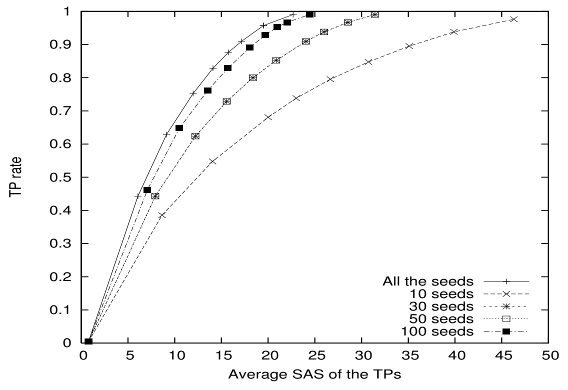
We took a second non-topological alignment pair from SCALI [16]. Figure 8 shows the non-topological alignment between 1FSF:A, and 1IG0:A. Our alignment had some common aligned segments with both SCALI and SARF2, but it returns a longer alignment. On the geometric SAS_3 measure STSA scored 1.27, SARF2 2.51 and SCALI 4.8. Among the sequential methods STRUCTAL was able to return a fairly good alignment for this pair, with a SAS_3 score of 1.6.

4. DISCUSSION

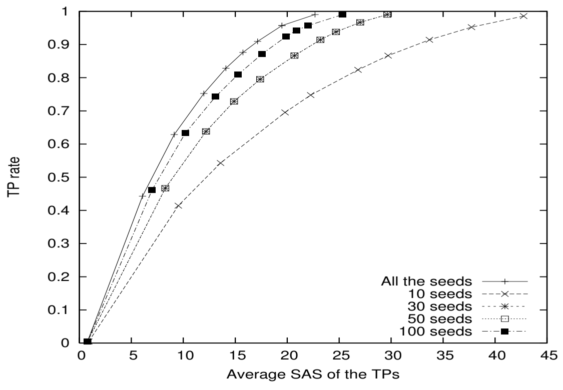
We presented STSA, an efficient algorithm for pairwise structural alignment. The STSA algorithm efficiently constructs an alignment from the superposed structures based on the spatial relationship between the residues. The algorithm assembles the alignment from closely superposed fragments, thus allowing for non-sequential alignments to be discovered.

Our approach follows a guided iterative search that starts from initial alignment seeds. We start the search from different initial seeds to explore different regions in the transformation search space.

On the challenging-to-align RIPC set [22], STSA alignments have higher agreement with the reference alignments than other methods: CE, DALI, FATCAT, MATRAS, CA, SHEBA, and SARF. The results on the RIPC set suggest that the STSA ap-

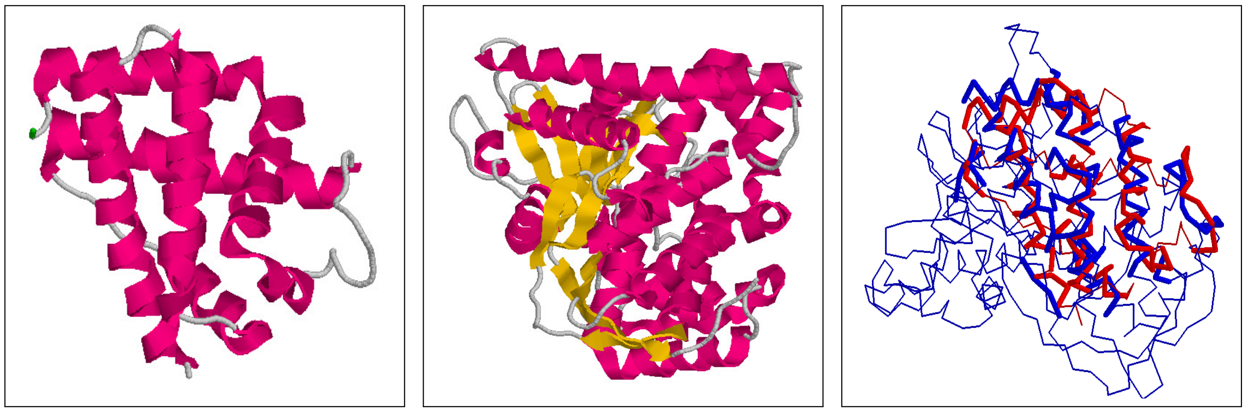


(a) Length score



(b) DALI rigid score

Fig. 6. Studying the effect of pruning on STSA. The average SAS score for the true positives as we consider different number of seeds is shown: (a) using length, (b) using DALI rigid score.



(a) (b) (c)

Fig. 7. A non-sequential alignment between (a) Leghemoglobin (2LH3:A, 153 residues) and (b) Cytochrome P450 BM-3 (2HPD:A, 471 residues). (c) STSA alignment: Leghemoglobin in black and Cytochrome in grey. The $N_{mat}/rmsd$ scores were 117/3.37Å for STSA, and 108/3.05Å for SARF2. For sequential methods, the scores were 56/2.27Å for STRUCTAL, 87/4.8Å for DALI and 91/4.05Å for CE.

proach is effective in finding non-sequential alignments, where the purely sequential (and in some cases non-sequential) approaches yield low agreement with the reference alignment.

Overall results on classifying the CATH singleton dataset show that STSA has high sensitivity for high specificity values. Moreover, the quality of STSA alignments, as judged by the SAS_3 geometric scores (longer alignments and lower $rmsd$), are better than the alignments of other methods: DALI, FAST, and STRUCTAL.

5. CONCLUSION & FUTURE WORK

Our experimental results on the RIPC set and the CATH dataset demonstrate that the STSA approach is efficient and competitive with state-of-the-art methods. Our next step is to extend our approach to address the multiple structure alignment problem. Moreover, we plan to add a functionality to handle flexible and reverse alignments.

References

1. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Res*, 5(28):235–242, 2000.
2. J.F. Gibrat, T. Madej, and S.H. Bryant. Surprising similarities in structure comparison. *Curr Opin Struct Biol*, 6: 377–385, 1996.
3. R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. *PNAS*, 101:12201–12206, 2004.
4. J. Xu, F. Jiao, and B. Berger. A parameterized algorithm for protein structure alignment. *J Comput Biol*, 5:564–77, 2007.
5. S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48:443–453, 1970.
6. S. Subbiah, D.V. Laurents, and M. Levitt. Structural similarity of dna-binding domains of bacteriophage repressors and the globin core,. *curr biol*, 3:141–148, 1993.

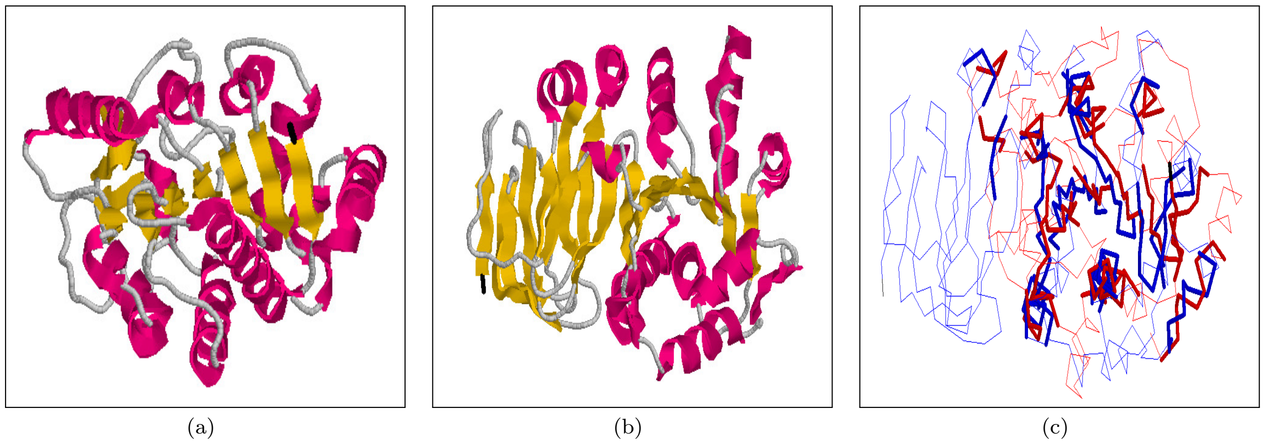


Fig. 8. A non-topological alignment between (a) Glucosamine-6-Phosphate Deaminase (1FSF:A, 266 residues) and (b) Thiamin Pyrophosphokinase (1IG0:A, 317 residues). (c) STSA alignment: 1FSF:A in black and 1IG0:A in grey. The $N_{mat}/rmsd$ scores were 139/3.42Å for STSA, 104/5.4Å for SCALI, and 105/2.9Å for SARF2. For the sequential methods the scores were 145/4.88Å for STRUCTAL, 106/4.9Å for DALI, and 111/5.1Å for CE.

7. M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. *Proc Int Conf Intell Syst Mol Biol*, 4: 59–67, 1996.
8. C.A. Orengo and W.R. Taylor. Ssap: sequential structure alignment program for protein structure comparison. *Methods Enzymol*, 266:617–35, 1996.
9. Y. Zhang and J. Skolnick. TM-align: A protein structure alignment algorithm based on TM-score.
10. M. Tyagi, V.S. Gowri, N. Srinivasan, A.G. Brevern, and B. Offmann. A substitution matrix for structural alphabet based on structural alignment of homologous proteins and its applications. *Proteins*, 65(1):32–39, 2006.
11. T. Can and T.F. Wang. Ctss:a robust and efficient method for protein structure alignment based on local geometrical and biological features. In *IEEE Computer Society Bioinformatics Conference (CSB)*, pages 169–179, 2003.
12. L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138, 1993.
13. N.N Alexandrov. Sarfing the pdb. *Protein Engineering*, 50 (9):727–732, 1996.
14. I.N. Shindyalov and P.E. Bourn. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein Eng*, 11:739–747, 1998.
15. J. Zhu and Z. Weng. Fast: A novel protein structure alignment algorithm. *Proteins:Structure, Function and Bioinformatics*, 14:417–423, 2005.
16. X. Yuan and C. Bystroff. Non-sequential structure-based alignments reveal topology-independent core packing arrangements in proteins. *Bioinformatics*, 21(7):1010–1019, 2003.
17. Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19:II246–II255, 2003.
18. I. Eidhammer, I. Jonassen, and W.R. Taylor. Structure comparison and structure patterns. *J Comput Biol*, 7(5):685–716, 2000.
19. M.R. Garey and D.S. Johnson. Computers and intractability: A guide to the theory of np-completeness. In *W.H. Freeman, San Francisco, CA*, 1979.
20. R. Nussinov and H.J. Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *proc. national academy of sciences of the usa (biophysics)*, 88:10495–10499, 1991.
21. F. Gao and M.J. Zaki. Indexing protein structures using suffix trees. In *IEEEComputational Systems Bioinformatics Conference, Palo Alto, CA*, 2005.
22. G. Mayr, F. Dominques, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Structural Biol*, 7 (50):564–77, 2007.
23. C.A. Orengo, A.D. Michie, S. Jones, D.T. Jones, M.B. Swindells, and J.M. Thornton. Cath- a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.
24. C. Bystroff, V. Thorsson, and D. Baker. Hmmstr: a hidden markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.*, 301:137–190, 2000.
25. W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr*, A32:922–923, 1976.
26. G.H. Golub and C.F. Van Loan. Matrix computations. *Johns Hopkins University Press*, 3, 1996.
27. M.I. Abouelhoda and E. Ohlebusch. Chaining algorithms for multiple genome comparison. *Journal of Discrete Algorithms*, 50(3):321–341, 2005.
28. R. Kolodny, P. Koehl, and M. Levitt. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J Mol Biol*, 346(4):1173–88, 2005.
29. A. Zemla. Lga - a method for finding 3d similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374, 2003.
30. J. Jung and B. Lee. Protein structure alignment using environmental profiles. *Protein Engineering*, 13:535–543, 2000.
31. M. Gerstein and M. Levitt. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *it Protein Sci*, 7:445–456, 1998.
32. A. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. Scop: A structural classification of proteins for the investigation of sequences and structures,. *J Mol Biol*, 247:536–540, 1995.