Journal of Bioinformatics and Computational Biology
© Imperial College Press

# ITERATIVE NON-SEQUENTIAL PROTEIN STRUCTURAL ALIGNMENT *

Saeed Salem[a], Mohammed J. Zaki[a †], and Christopher Bystroff[a,b]
*[a] Department of Computer Science, [b] Department of Biology, Rensselaer Polytechnic Institute,*
*110 8th st. Troy, New York 12180, USA*
*Email: {salems, zaki, bystrc}@cs.rpi.edu*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Structural similarity between proteins gives us insights into their evolutionary relationships when there is low sequence similarity. In this paper, we present a novel approach called SNAP for non-sequential pair-wise structural alignment. Starting from an initial alignment, our approach iterates over a two-step process consisting of a superposition step and an alignment step, until convergence. We propose a novel greedy algorithm to construct both sequential and non-sequential alignments. The quality of SNAP alignments were assessed by comparing against the manually curated reference alignments in the challenging SISY and RIPC datasets. Moreover, when applied to a dataset of 4410 protein pairs selected from the CATH database, SNAP produced longer alignments with lower *rmsd* than several state-of-the-art alignment methods. Classification of folds using SNAP alignments was both highly sensitive and highly selective. The SNAP software along with the datasets will be made available online at `http://www.cs.rpi.edu/~zaki/software/SNAP`.

*Keywords*: protein structure alignment; pairwise alignment; non-sequential alignment; iterative alignment

## 1. INTRODUCTION

Over the past years, the number of known protein structures has been increasing at a relatively fast pace, thanks to advancement in NMR spectroscopy and X-ray crystallography. Recently (as of June 2008) the number of protein structures in the Protein Data Bank(PDB) [1] has reached 47526. 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

[*] This work was supported in part by NSF Grants EMT-0829835, and CNS-0103708, and NIH Grant 1R01EB0080161-01A1.
[†] Corresponding author.

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 discretizing 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 discretization techniques. Despite the polynomial time approximate algorithms and as their authors 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 pairwise structural alignment problem. Over the years, a number of heuristic approaches have been proposed, which can mainly be classified into two main categories, dynamic programming and clustering.

### 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 matching 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 hexa-peptide 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 $\alpha$-helix or $\beta$-strand over the $C_\alpha$ 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 w [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 SNAP[a], an efficient non-sequential pair-wise structural alignment algorithm. SNAP 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, our greedy approach allows for non-topological (non-sequential) similarity to be extracted.

The SNAP approach is a two-step process. First, we compile a list of alignment seeds, also called aligned fragment pairs (AFPs) in some methods. After extracting seeds, we generate an alignment for each seed and report the alignment which has the best score. For the first step, we use two sources for the seeds. We employ

[a] a non-sequential permutation of the bold letters in **N**on-sequential **P**rotein **S**tructure **A**lignment

PSIST [21] to generate a list of similar substructures. A second source of seeds is the HMMSTR method [22]. We discuss in details how these two methods work in the next section.

To assess the quality of the SNAP alignments, we tested it on the recently published hard-to-align SISY, and RIPC sets [23]. SNAP alignments have higher agreement (accuracy) with the reference alignments than the agreements of the state-of-the-art methods like CE, DALI, FATCAT, MATRAS, CA, SHEBA, SARF, LGA, and SCALI, on the RIPC dataset, and it has the second best accuracy (after DALI) on the SISY dataset.

Yuan and Bystroff [16] have pointed out that an increased length and lower *rmsd* does not necessarily signal a more biologically meaningful alignment. Errors such as disjoint subgraphs, local structural mismatches, and aligning paired beta strands to unpaired strands, were commonly found in automated alignments, except SCALI. To avoid this problem, we have made all our comparisons with a gold standard that is manually curated. Unlike for SCALI, no measures were taken to avoid specific types of alignment error in SNAP. Nonetheless, the results with respect to the curated reference alignments suggest that non-biological errors have been minimized.

We also compiled a dataset of 4410 protein pairs from the CATH [24] to asses how SNAP performs as a classifier of protein topology. To predict whether two proteins have the same topology classification, we simply aligned the structures and applied a cutoff on the alignment score. Results from the CATH dataset indicate that SNAP achieves high sensitivity and selectivity levels and is competitive to well established structure comparison methods like DALI, STRUCTAL, and FAST, as judged by their ROC curves. Moreover, we show that SNAP alignments which are labeled as true positives have a better average geometric score than alignments of an equally sized set of alignments produced by the different methods.

## 2. ALIGNMENT SEEDS

Our approach is based on finding an alignment using initial seeds. In this section we discuss two methods for getting the initial alignment seeds and we present our greedy chaining algorithm in the next section.

### 2.1. *PSIST seeds*

The initial alignment seeds are similar substructures between protein $A$ and protein $B$. An initial seed is an equivalence between a set of residue 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).

### 2.1.1. *Mapping protein structure to a sequence of structural alphabet*

Let $P = \{a_1, a_2, \ldots, a_n\}$ represent a protein, where $a_i$ is the $i$th-residue, along the backbone, represented by the coordinates of the $C_\alpha$ atoms. The structure-feature sequence (*SF-sequence*) of the protein $P$ is defined as $P^s = \{p_1^s, p_2^s \ldots p_{n-w+1}^s\}$, where $p_i^s$ is the $i$-th normalized feature vector representing the $i$-th residue and $w$ is the window size.

Each feature vector captures the local geometry of the corresponding residue within a sliding window of size $w$ along the backbone of the protein. The feature vector $p_i^v$ is composed of the distances and the dihedral angles between the first residue $i$ and all the other residues $j$ ($j \in [i+1, i+w-1]$) within the window:

$$p_i^v = \{d_{i,i+1}, \cos(\theta_{i,i+1}), \ldots, d_{i,i+w-1}, \cos(\theta_{i,i+w-1})\}$$

where $d_{i,j}$ is the distance between the $C_\alpha$ atoms of residues $p_i$ and $p_j$, and $\theta_{i,j}$ gives the dihedral angle between the two planes defined over the $C, N, C_\alpha$ atoms of residues $a_i$ and $a_j$, respectively. With a window of size $w$, the dimension of $p_i^v$ is $k = 2 * (w-1)$. To reduce the number of possible feature vectors, we normalize $p_i^v$ to get the normalized feature vector $p_i^s = \{n_1, n_2, \ldots, n_k\}$, where $n_i$ is an integer within the range $[0, b-1]$.

For simplicity of representation, we treat each $p_i^s$ as base-$b$ number (or symbol) with $k$ digits, giving us at most $b^k$ different symbols comprising the new structural alphabet $\Sigma$. For a given protein $P = \{a_1, a_2, \ldots, a_n\}$, the structure-feature sequence *SF-sequence* of the protein is defined as $P^s = \{p_1^s, p_2^s \ldots p_{n-w+1}^s\}$, where $p_i^s$ is the alphabet representing the normalized feature vector of the $i$-th residue. To summarize, the *SF-sequence* $P^s$ is a sequence of size $n - (w+1)$ over the structural alphabet $\Sigma$ [21].

### 2.1.2. *Finding maximal similar subsequences*

The problem of finding similar substructures is mapped to finding similar SF-subsequences, by virtue of the structural alphabets. We use suffix tree indexing for finding the maximal subsequences [25]. Suffix trees can be constructed in linear time and once constructed many string problems can be solved in linear or constant time [26].

To find all maximal matches between the query SF-sequence $Q^s$ and the template SF-sequence $P^s$ we build two suffix trees, $GST_q$ and $GST_d$. We then traverse the two suffix trees simultaneously to retrieve all the maximal matches. The result is the set of maximal matching segments (MMSs) for proteins $A$ and $B$, $S = \{F_i^A F_j^B(l)\}$, where each MMS is composed of a fragment of protein $A$ starting at residue $i$ and a fragment of protein $B$ starting at residue $j$ and the two fragment have the same length, $l$. For the full algorithm, please refer to [21].

### 2.2. *HMMSTR seeds*

We used the HMMSTR [22] method to identify the alignment seeds following the same approach used in SCALI [16]. This allows us to compare our proposed method for alignment propagation directly with the fragment assembly method proposed in SCALI which uses the same set of HMMSTR fragments. Each seed is an ungapped alignment of two segments of any length greater than 4, as long as they have no backbone angle deviations greater than 120°. These fragments are scored according to sequence similarity as measured using HMMSTR, a hidden Markov model for local sequence/structure correlations in proteins.

The sequence score is the sum over the aligned positions of the joint probabilities of HMMSTR Markov states, as follows: $g = \sum_{i=1,L}(\sum_{q=1,282}(\gamma_{qi}^{A} \times \gamma_{qi}^{B}))$, where $\gamma_{qi}^{A}$ is the a posterior probability of HMMSTR state $q$ at position $i$ in the fragment, given the sequence of the protein $A$. HMMSTR predicts local structure, thus $g$ measures the similarity between structure predictions. But $g$ also enforces the alignment of similar sequence patterns. For example, two helices may not align with the polar side of one helix aligned with the non-polar side of the other, even though the local structure may align perfectly. The SCALI method [16] constructs the alignment by clustering the fragments using a near-greedy approach. An assembly of aligned fragment is scored using $g$ plus a contact map alignment score plus a score penalizing non-sequential breaks in the alignment. There is no penalty for sequential gaps. The 100 best alignments are kept at each stage. In a final refinement step, the fragments are shortened or extended based on *rmsd*. The set of HMMSTR seeds consists of fragments, $S = \{F_i^A F_j^B(l)\}$, as in the case of PSIST seeds.

## 3. SNAP ALIGNMENT

### 3.1. *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.

#### 3.1.1. *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 [27, 28].

### 3.1.2. *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 $\delta$, and 0 otherwise.

### 3.1.3. *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 matching pairs 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, $R_k$, is a set of consecutive diagonal 1's in the scoring matrix $S$. A run constitutes an equivalence, between a substructure in $A$ and another in $B$, that can be superposed with a small *rmsd*. Specifically, a run $R_k$ 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 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 $R = \{R_1, R_2, \cdots, R_{|R|}\}$ 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.

Two runs $R_i$ and $R_j$ can be joined if $Ri \prec Rj$ ($Ri$ precedes $Rj$) or $R_j \prec R_i$ in the scoring matrix $S$. For strictly sequential alignment, the precedence relation is transitive which means that if $R_i \prec R_j$ and $R_j \prec R_k$, then $R_i \prec R_k$. Constructing the optimal (longest) sequential chain of runs $C \subseteq R$ can be solved efficiently [26].

Unfortunately, the case is not the same for non-sequential alignment. The main reason is that the precedence relation is not transitive which means that if $R_i \prec R_j$ and $R_j \prec R_k$, then it is not guaranteed that $R_i \prec R_k$. For the general problem of selecting the longest chain of runs, which can have runs chained non-sequentially, we have to ensure that every pair of runs chosen in $C$ are non-overlapping. Therefore, the general problem of finding the subset of runs with the largest length is essentially the same as finding the maximum weighted clique in a graph $G = (V, E, w)$ where the set of vertices $V$ represents the set of runs, each vertex $v_i$ has a weight given as $w(v_i) = |R_i|$, and there is an edge $(v_i, v_j) \in E$ if the runs $R_i$ and $R_j$ do not overlap (can be joined together).

The problem of selecting the maximum non-overlapping runs maps to finding the maximum weighted clique in a graph because we have to ensure that every pair of runs 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 we can construct the longest sequential chain of runs in an efficient way [26]. However, since we are interested in non-sequential alignments whose solution is computationally expensive, we adopt the greedy approach for finding the maximum weighted clique.

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 [29]. 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 $E_1$.

### 3.1.4. *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: $\delta$ 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.

### 3.2. *Scoring the alignments*

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

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

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 on the quality of the alignment. 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. [30] recently contended that for some alignment methods, scoring the alignment by geometric measures yields better specificity and sensitivity; we observe consistent behavior in our results.

### 3.3. *Handling reverse alignments*

Reverse alignments capture the cases where helices or beta strands reverse direction. Both secondary structure types can pack well and make the right types of energetic interactions when the chain direction is reversed, so for the purposes of structure prediction a reverse alignment might be biologically relevant.

If we consider anti-diagonal runs in our scoring matrix, then the SNAP algorithm will find alignments that have reverse segments. An anti-diagonal run is a triplet $((a_i, b_j, l)$, where the correspondence between residues is as follows: $\{(a_i, b_j), (a_{i+1}, b_{j-1}), \cdots, (a_{i+l-1}, b_{j-l+1})\}$.

### 3.4. *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, for PSIST seeds we select the most promising seeds based on two heuristics: first, the length of the seed; second, the DALI rigid similarity score [12]. For the HMMSTR seeds, we select the seeds based on HMMSTR sequence score. 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 by selecting fewer seeds.

### 3.5. *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 $O(n^2)$.

### 3.6. *An Alignment example*

Before we go into the results section, we want to show an example of how the algorithm finds the alignment given an alignment seed. Figure 2 shows the alignment between the SH3 domain (PDB code 1aww_, 67 residues) and PsaE subunit (PDB code 1gxiE, 73 residues). Superposing the structures based on the transformation obtained by optimally superposing the initial seed, $E_0 = \{(11, 8), (12, 9), \cdots, (16, 13)\}$, we get the set of runs shown in Figure 2(a), from which we select the set of the runs, shown in bold. The selected set of runs makes up the alignment which we use to find a superposition and get a new alignment. This process of superposition and alignment is repeated until a maximum number of iterations is reached or the alignment can not be refined, i.e. the score does not improve. The final alignment $E_3$ has a 100% agreement with the reference manually curated alignment provided in the SISY dataset [23].

## 4. RESULTS & DISCUSSION

To assess the quality of SNAP alignments compared to other structural alignment methods, we tested our method on the hard-to-align SISY and RIPC sets [23]. To evaluate the overall sensitivity and specificity of SNAP compared to other alignment methods, we looked at 4410 alignment pairs from the CATH [24] as a gold standard for classification.

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], LGA [31], SCALI [16], and FAST [15]. For the SISY and RIPC datasets, we used the published results for CE [14], FATCAT [17], CA [32], MATRAS [33], and SHEBA [34].

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 SNAP were $r = 3$, $\delta = 4.5\mathring{A}$ and using top 200 initial seeds (see Section 4.4 for more details).

### 4.1. *SISY set*

The SISY dataset is a subset of SISYPHUS, a database of non-trivial structural alignments that include proteins with circular permutations, segment-swapping, context-dependent folding or chameleon sequences that can adopt alternative secondary structures [35]. For each multiple structure alignment in SISYPHUS, the pair with the lowest identity was included in the SISY dataset. The selected pairs were later pruned such that no pair has a sequence identity more than 40% and that no structure has more than one chain. The reference alignments for the SISY dataset were extracted from the SISYPHUS database as well. The final set of alignments in the SISY dataset included 69 structure pairs. Among them, 52 structure pairs are categorized as homologous in SISYPHUS, while the remaining 17 pairs are related through a common fold or a fragment definition.

We measured the agreements of the alignments of different methods with the reference alignments provided in the SISYPHUS database, which are manually curated. The agreement of a given alignment with the reference alignment is defined as the percentage of the residue pairs aligned identically to the reference alignment($I_s$) relative to the reference alignment's length ($L_{ref}$).

Figure 3 shows the distribution of the percentage of agreement for different alignment methods on the SISY dataset. DALI has the highest mean accuracy of 76% followed by SNAP with mean accuracy of 73%; all the remaining methods have a mean accuracy less than 68%, with the CA method having the lowest mean accuracy of 51%. For the alignment pairs defined as fragment in SISYPHS, it was hard to get a high accuracy all the time since SISYPHUS defines a short reference alignment and the methods seek a larger global alignment. Both DALI and SNAP had high accuracy on some of these pairs, while all the other methods had zero agreement.

### 4.2. *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 [23], we compute the agreement between an alignment $s$ and the reference alignment $ref$ as the percentage of the residues aligned identically to the reference alignment($I_s$) relative to the reference alignment's length ($L_{ref}$), $agreement(s, ref) = I_s/L_{ref}$.

As shown in Figure 4, while all the methods (except SCALI) have mean agreements equal to 60 percent or lower, the mean agreement of SNAP alignments is 71%, and 64% for SCALI. As for the median, all the methods except FATCAT (63%) and SCALI (69%) have median agreements less than 60%, while SNAP alignments have a median agreement of 67% .

As Mayr et al. [23] noted, there are seven challenging protein pairs which reveal how repetition, extensive indels, circular permutation, and conformational changes result in low agreements 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 SNAP. First, for alignment of L-2-Haloacid dehalogenase (PDB code 1qq5, chain A, 245 residues) with CheY protein (3chy, 128 residues), all the methods except SARF (33%), and SCALI (66%) returned zero agreement with the reference alignment while SNAP returned 100% agreement. This pair is hard to align because it has a circular permutation and an insertion. The second problematic pair was of the alignment of NK-lysin (1nkl, 78 residues) with prophytepsin (1qdm, chain A, 77 residues) and it has a circular permutation. For the second pair, most methods (except CA returned 41%, SARF returned 92%, and SCALI returned 69%) returned zero agreement with the reference alignment while SNAP returned 99 percent agreement. In this pair the N-terminal region of domain 1nkl has to be aligned with the C-terminal region of domain 1qdm to produce an alignment that matches the reference alignment (see Figure 5). 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 SNAP alignments with the reference alignments are higher than the agreements of CA, SARF, and SCALI. As shown in Figure 5, 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. SCALI also misses 31% of the reference alignment pairs. One the other hand, both SARF and SNAP alignments have excellent agreement with the reference alignment, 92%, 99%, respectively.

Our proposed approach is not designed to handle flexible alignments and thus its agreements with the reference alignments for the pairs which have conformational change are not high, compared to the methods which can handle flexible alignments. FATCAT has the best agreements on the pairs with conformational change since it allows for flexible alignments by introducing twists in the structures. However, FATCAT has an inherent limitation of producing only sequential alignments due to the way it chains the set of alignment fragment pairs (AFPs) and thus has lower agreements on the pairs with circular permutation.

### 4.3. *Evaluation of classification of the CATH*

Gerstein and Levitt [36] emphasized the importance of assessing the quality and significance of structural alignment methods using an objective approach. They used the SCOP database [37] as a gold standard to assess the sensitivity of the structural alignment program against a set of 2107 pairs that have the same SCOP superfamily. In a more recent work, Kolodny et al. [30] presented a comprehensive comparison of six protein structural alignment methods. They used the CATH classification [24] 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 true and false positive rates of SNAP alignments to those of other three methods: DALI, STRUCTAL, and FAST. Since the other methods report only sequential alignments, for SNAP we restrict the greedy algorithm to report only sequential alignments.

#### 4.3.1. *The CATH Singleton Dataset*

CATH [24] is a hierarchical classification of protein domain clusters. The CATH database clusters structures using automatic and manual methods. The CATH database (version 3.1.0; Jan'07) 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 $(10 \times 21)$ domains selected from the different sibling superfamilies. Our final dataset thus has 4410 alignment pairs $(21 \times 210)$. 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.

#### 4.3.2. ***Alignment Results***

We ran all the methods on the 4410 structure pairs. All methods report the number of residues in the alignment, the *rmsd* of the alignment, and the native score of the alignment: STRUCTAL reports a $p$-value score for the alignment, FAST reports a normalized score (SN), and DALI reports a $z$-score. For SNAP, we score the alignments using the geometric matching score $SAS_3$. For each method, we sort the alignments by the method's native score and vary a threshold $k$. Then we 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 $k$ scoring pairs. Moreover, we compare the quality of the alignments of different methods by comparing the average $SAS_3$ geometric score for the true positives.

Figure 6(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. The method that has the uppermost ROC (largest area under the curve) is the one that agrees most with the gold standard. Dali performed the best with area 0.88; STRUCTAL came second with 0.87, then SNAP with 0.85, and last comes Fast with 0.80. By zooming in on the ROC curve, Figure 6(b) shows that SNAP has competitive sensitivities at low false positive rates.

Having the best ROC curve does not imply the best alignments. Kolodny et al. [30] showed that the best methods, with respect to the ROC curves, do not necessarily have the best average geometric score for the true positives. Two methods can order the structure pairs similarly while one of the methods reports better alignments all the time. Our results confirm this observation. Figure 6(c) shows the average $SAS_3$ measure of the true positives for different sensitivity values. This figure compares the average score for an equal number of alignments produced by the different methods involved.

SNAP has a better average $SAS_3$ score for the true positives for the first half of the graph, then STRUCTAL becomes better. This can be explained by the fact that we use the $SAS_3$ measure in our algorithm. While it is able to classify as many true positives as the other methods, FAST has the worst average $SAS_3$ measure.

These results suggest that it is possible to successfully discriminate between fold classes without getting the alignment right. Although counter-intuitive, it is a well-known phenomenon in the field of remote homology detection by sequence alignment that good recognition accuracy does not imply good alignment accuracy [38]. Moreover, for structure pairs which are related through a conformational change, a score that is solely based on rigid body transformation will not be able to capture all the true positive pairs. Nevertheless, we use the $SAS$ score to assess the geometric quality of the alignment produced by the different methods.

Figure 6(d) shows the ROC curve of all the methods after sorting the alignments based on the geometric match score, $SAS_3$. A lower geometric score corresponds to a better alignment. There is a variation in the performance of the alignment methods when we use the geometric score to sort the alignments. The area under the ROC curve for each of the three methods decreases as compared to the area under the curve produced when the native score is used for sorting the alignments. The area for STRUCTAL decreases from 0.87 to 0.80, for FAST it decreases from 0.80 to 0.78, for DALI it decreases from 0.88 to 0.81, and for SNAP it stays 0.85. While the methods seem to agree with the gold standard when the alignments are sorted by the native score, their performance slightly decreases when we sort the alignments by the geometric score. Specifically, the alignments do not have the best geometric score, and the ROC curve is lower.

### 4.3.3. *Running times*

Table 4.3.3 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. SNAP was the second fastest with 1719 seconds, STRUCTAL came third, and DALI was the slowest. Of the 1719 seconds taken by SNAP, 68 seconds were taken to get the seeds using PSIST and the remaining 1651 seconds to generate the alignments. For SNAP alignments, we used only the top (longest) 200 seeds.

## 4.4. ***Analysis of* SNAP**

There are some parameters that affect the quality of the alignment in SNAP, namely $L$ the maximum *rmsd*, $r$ the minimum length of the run, $\delta$ the threshold distance which is used to populate the scoring matrix, the number of initial seeds, and lastly $k$ used in $SAS_k$. The optimal values for $L = 4.5$, $r = 3$, and $\delta = 4.5$ were found empirically such that they give the best ROC curve on the CATH dataset. First we study the effect of seeds pruning by selecting the longest PSIST seeds. Figure 7(a) shows that as we consider more seeds we get better ROC curves. When we consider the 200 longest seeds, we get the same area under the curve as the case when we consider all the seeds. However, by considering all the seeds we get higher true positive rates for low false positive rates as show in Figure 7(b). Seeds pruning results in a drastic reduction in the running time of SNAP: it takes 5740 seconds to run on all the seeds while it takes 1719 second for the 200 seeds and 595 seconds for the 50 top seeds.

Second, we investigate how the ROC curve changes when we use different $k$ to calculate the geometric score. Figure 7(c) shows that for $k = 1$ and $k = 2$ the performance of SNAP is drastically affected with areas under the ROC curve 0.49 and 0.77 respectively. On the other hand the area under the curve mostly stays the same (0.85) for $k = 3, 4$, and 5. For $SAS_1$ and $SAS_2$, short alignments with small *rmsd* will get good score and their ranking will be high and thus will be labeled as true positives even though they are not and that explains why their ROC curves are not as good.

Next, we investigate the effect of using different sources for the seeds. Figure 7(d) shows that using all or the top 200 PSIST seeds gives a similar ROC curve to using all the HMMSTR seeds. However, in the high selectivity region (low false positive rates), considering all the HMMSTR seeds results in higher true positive rates. The quality of HMMSTR seeds comes at a higher running time. HHMMSTR takes 11507 seconds to extract the seeds while PSIST takes only 68 seconds.

### 4.5. *Two non-sequential alignments*

To demonstrate the quality of SNAP in finding non-sequential alignments, we show SNAP alignments on two non-sequential alignment pairs presented in earlier methods, SARF2 [13], and SCALI [16].

Figure 8 shows a non-sequential alignment between Leghemoglobin (2LH3:A) and Cytochrome P450 BM-3 (2HPD:A). SNAP and SARF2 has some common aligned segments, but SNAP yielded an alignment of length 118 and $rmsd = 3.37\mathring{A}$, whereas SARF2 yielded an alignment with length 108 and $rmsd = 3.05\mathring{A}$. The $SAS_3$ score of SNAP is 2.05, 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 9 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, length is 127. On the geometric $SAS_3$ measure, the score were 1.7 for SNAP, 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.

## 5. CONCLUSIONS

We presented SNAP, an efficient algorithm for pair-wise protein structural alignment. The SNAP 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 [23], SNAP alignments have higher agreements with the reference alignments than the other methods: CE, DALI, FATCAT, MATRAS, CA, SHEBA, SCALI, and SARF. The results on the RIPC set suggest that the SNAP approach 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. Also on the SISY set [23], SNAP has competitive agreements with the reference alignments where it comes second after DALI, the best method on the SISY set. The overall results on classifying the CATH singleton dataset show that SNAP has high sensitivities for low false positive rates. Moreover, the quality of SNAP alignments, as judged by the $SAS_3$ geometric scores, are better than the alignments of other methods: DALI, FAST, and STRUCTAL.

One obvious next step is to extend our approach to address the multiple structure alignment problem. In addition, we plan to add a functionality to handle flexible 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.
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. *Nucleic Acids Research*, 33:2302–2309, 2005.
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. 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.
23. G. Mayr, F. Dominques, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Structural Biol*, 7(50):564–77, 2007.
24. 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.
25. E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.
26. D. Gusfield. *Algorithms on strings, trees, and sequences: Computer science and computational biology.* 1999.
27. W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr*, A32: 922–923, 1976.
28. G.H. Golub and C.F. Van Loan. Matrix computations. *Johns Hopkins University Press*, 3, 1996.
29. M.I. Abouelhoda and E. Ohlebusch. Chaining algorithms for multiple genome comparison. *Journal*

*of Discrete Algorithms*, 50(3):321–341, 2005.

30. 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.
31. A. Zemla. Lga - a method for finding 3d similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374, 2003.
32. O. Bachar, D. Fischer, R. Nussinov, and H.J. Wolfson. A computer vision based technique for 3-d sequence independent structural comparison of proteins. *Protein Engineering*, 6(3):279–288, 1993.
33. T. Kawabata. Matras: a program for protein 3d structure compariso. *Nucleic Acids Res.*, 31:3367–9, 2003.
34. J. Jung and B. Lee. Protein structure alignment using environmental profiles. *Protein Engineering*, 13:535–543, 2000.
35. A. Andreeva, A. Prlic, T.J.P. Hubbard, and A. Murzin. Sisyphus - structural alignments for proteins with non-trivial relationships. *Nucleic Acid Research Database Issue*, 35:D253–D259, 2007.
36. 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.
37. 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.
38. L. Jaroszewski, W. Li, and A. Godzik. In search for more accurate alignments in the twilight zone. *Protein Science*, 11(7):1702, 2002.
39. I. Michalopoulos, G.M. Torrance, D.R. Gilbert, and D.R. Westhead. TOPS: an enhanced database of protein structural topology. *Nucleic Acids Research*, 32(Database Issue):D251, 2004.

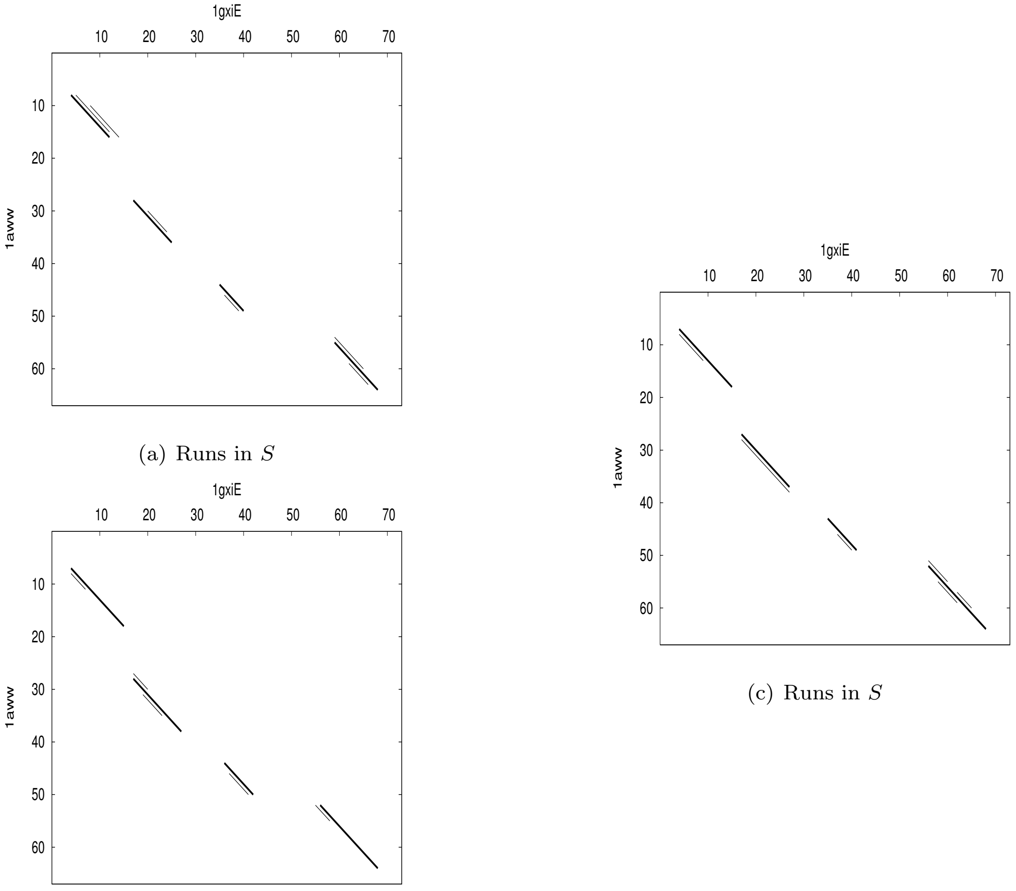
```
$M = \{F_i^A F_j^B(l)\}$, set of seed alignments
$L$, the *rmsd* threshold
$r$, the min threshold for the length of a run in $S$
$\delta$, the max distance threshold for $S$

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 SNAP Algorithm

Table 1. Comparison of the running times (in seconds) on the CATH Dataset.

| Method | DALI | STRUCTAL | FAST | SNAP |
|---|---|---|---|---|
| Time | 5532 | 3162 | 224 | 1719 |



(a) Runs in $S$

(b) Runs in $S$

(c) Runs in $S$

Fig. 2. An example to demonstrate the process of finding a global pair-wise structural for the two proteins, 1aww_, 67 residues and 1gxiE, 73 residues, based on the initial equivalence: $E_0 = \{(11, 8), (12, 9), \cdots, (16, 13)\}$. Only the runs that exceeded the length threshold are shown in the figure. (a) The initial runs shown here are the runs obtained from the scoring matrix $S_0$ that is obtained using the optimal superposition of the initial equivalence, $E_0$; using our greedy approach, we select a set of runs (shown in bold) which make up the alignment $E_1$: 55-64/59-68 8-16/4-12 28-36/17-25 44-49/35-40, runs are sorted by decreasing length; $|E_1| = 34$, *rmsd*= 2.2, and the geometric match score $SAS_3 = 56.13$. (b) The two structures are superposed based on $E_1$; the set of runs are shown here, and the selected runs define the new alignment $E_2$:52-64/56-68 7-18/4-15 27-37/17-27 43-49/35-41; $|E_2| = 43$, *rmsd*= 2.39, and the geometric match score $SAS_3 = 30.06$. (c) To refine the alignment further, we use $E_2$ to obtain the runs shown here, from which we select the runs that make up the refined alignment, $E_3$: 52-64/56-68 7-18/4-15 28-38/17-27 44-50/36-42. $|E_3| = 43$, *rmsd*= 1.84, and $SAS_3 = 23.15$. The alignment in $E_3$ can not be refined any further, thus we stop our iterative process of superposition and alignment.

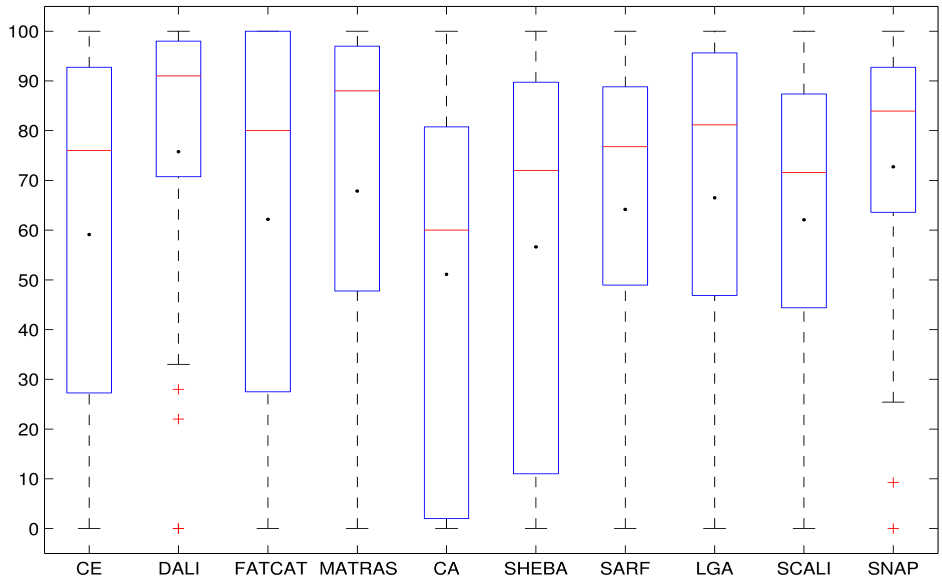


Fig. 3. Comparison of the alignments of 10 methods with the reference alignments from the SISY set. Box-and-whisker plots for the distribution of agreements of the alignments produced by different methods as compared to the true reference alignments. The dark dots indicate the means, the red horizontal lines indicate the medians, and the box shows the range between the lower and the upper quartiles. Results for LGA, SARF2, SCALI, and SNAP were obtained in-house while the results for the remaining methods were taken from [23].

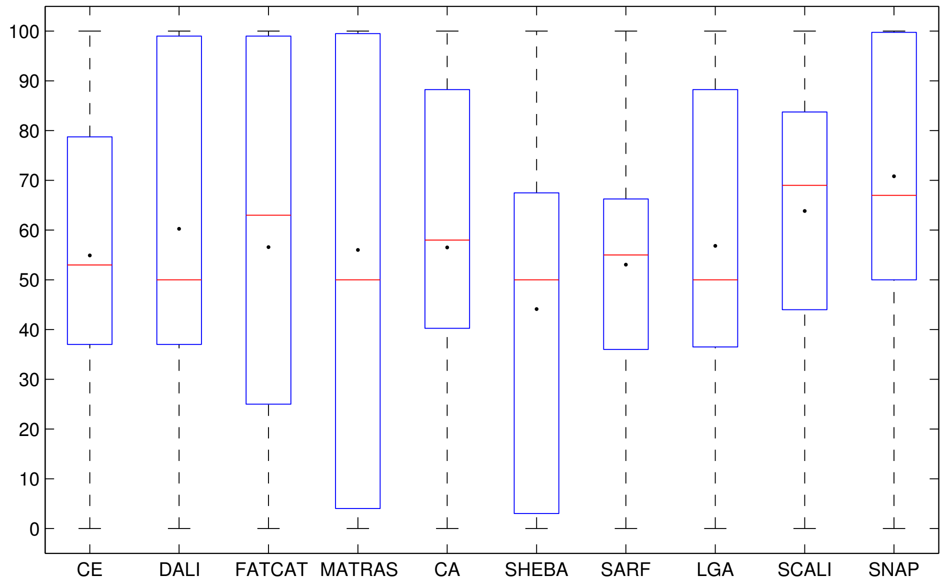


Fig. 4. Comparison of the alignments of 10 methods with the reference alignments from the RIPC set. Box-and-whisker plots for the distribution of agreements of the alignments produced by different methods as compared to the true reference alignments. The dark dots indicate the means, the red horizontal lines indicate the medians, and the box shows the range between the lower and the upper quartiles. Results for LGA, SARF2, SCALI, and SNAP were obtained in-house while the results for the remaining methods were taken from [23].

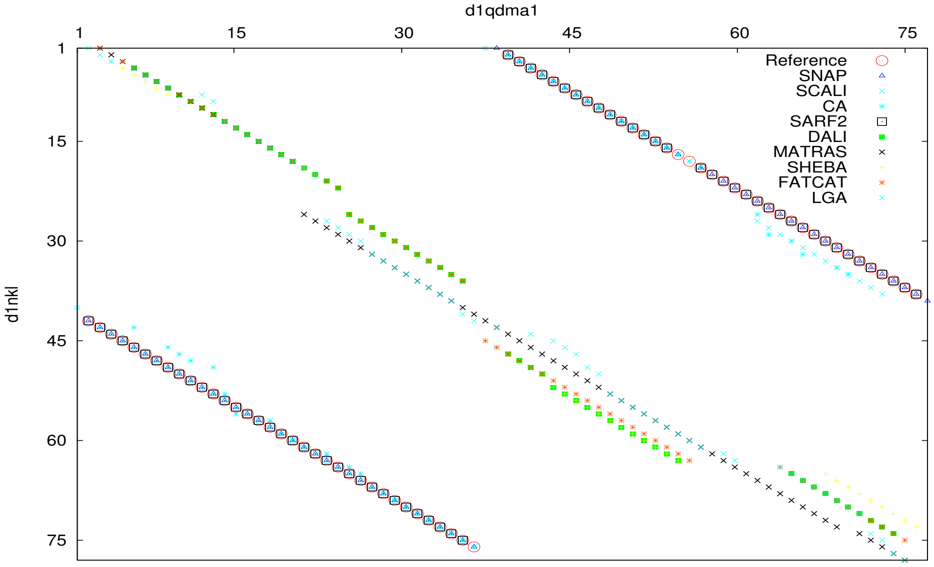
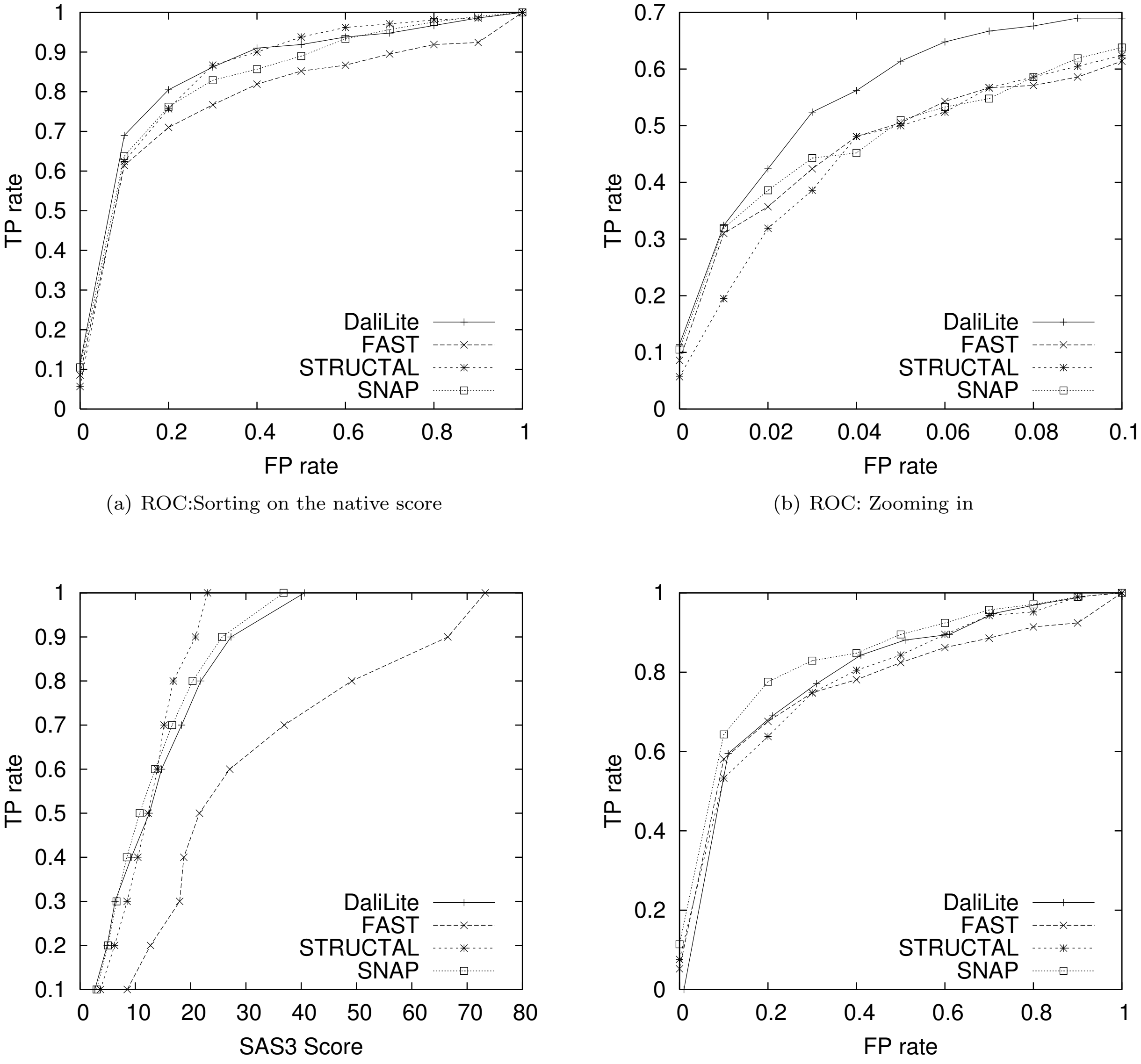


Fig. 5. Comparison of the agreement with the reference alignment of SNAP alignment and 6 other alignment methods. Residue positions of d1qdma and d1nkl__ are plotted on the x-axis and y-axis, respectively. Note: The reference alignment pairs are shown in circles. The CA, SARF, SCALI, and SNAP plots overlap 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.



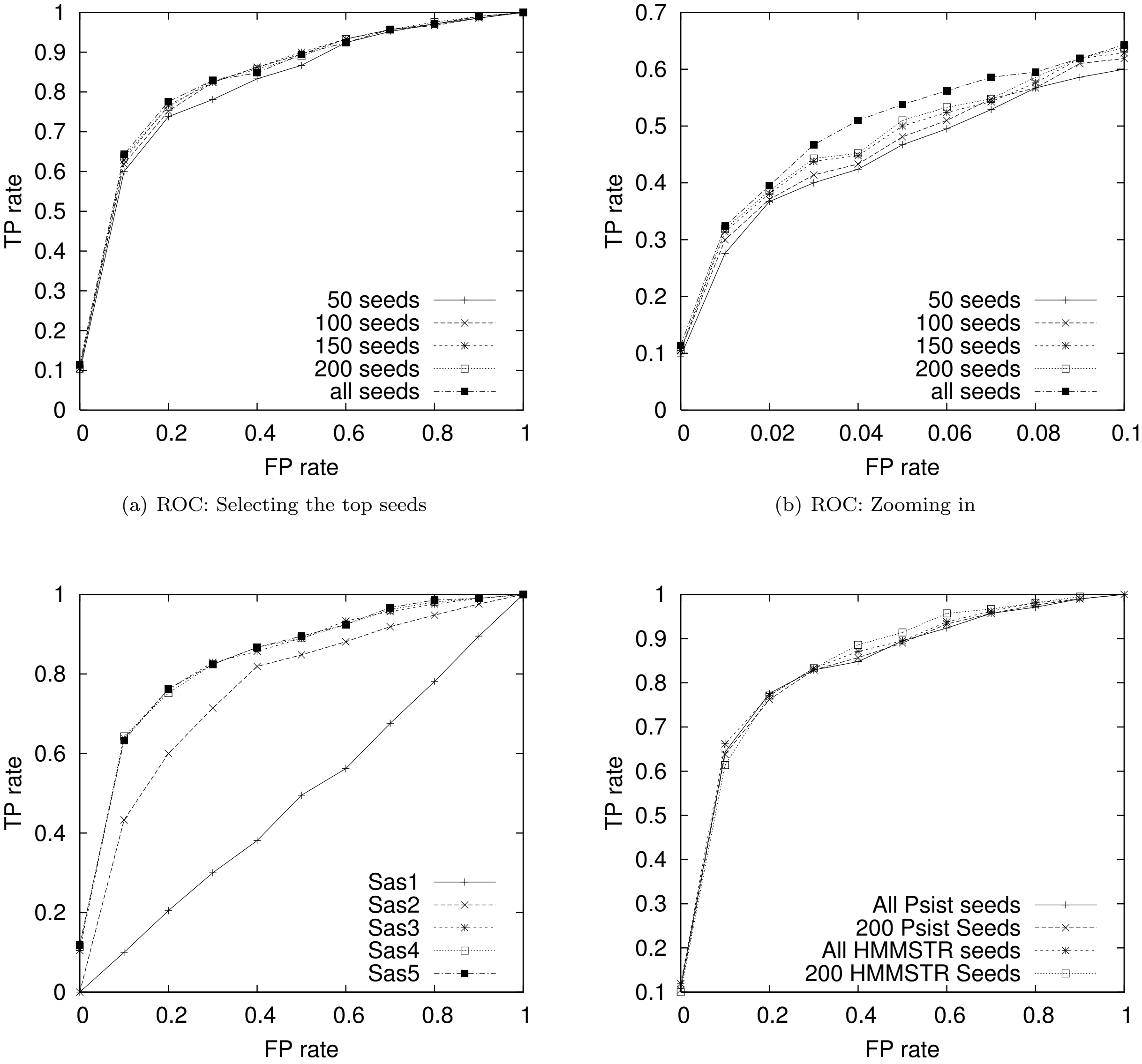
(a) ROC:Sorting on the native score

(b) ROC: Zooming in

(c) Average $SAS_3$ for the true positives

(d) ROC: Sorting on the $SAS_3$ score

Fig. 6. Receiver Operating Characteristic (ROC) curves for the structural alignment methods measured over the 4410 CATH pairs. (a) The alignments are sorted based on the native score or on the geometric match measure $SAS_3$. We tallied the number of true positives and false positives using CATH as a gold standard. (b) A zoom-in on the range of true positive rates from 0 to 0.1. (c) The average $SAS_3$ scores versus the true positive rate. (d) For all the methods, the alignments are sorted using $SAS_3$ scores and we plot the ROC curve showing the true positive rate vs the false positive rate.



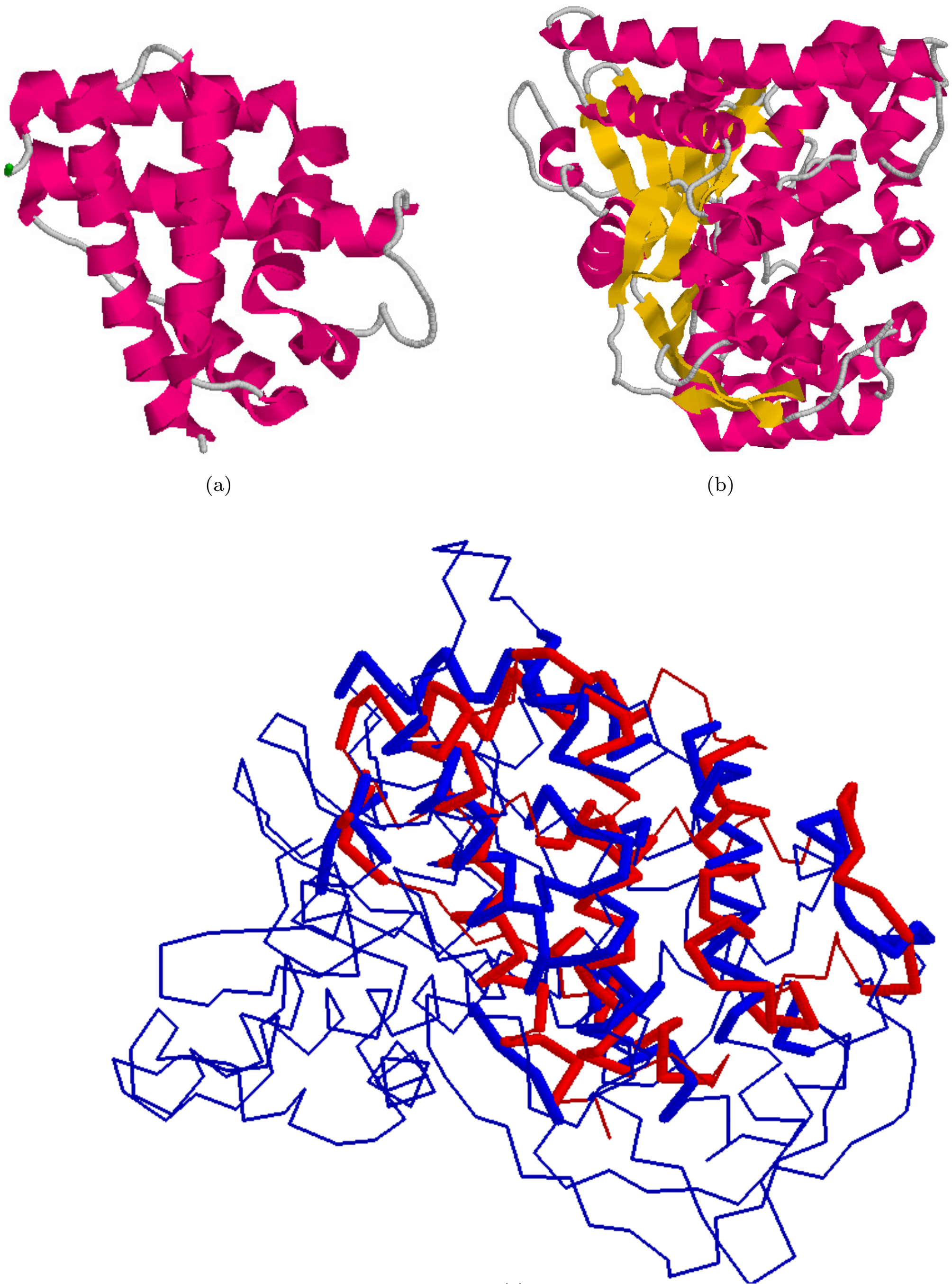
(a) ROC: Selecting the top seeds

(b) ROC: Zooming in

(c) ROC: SNAP using different $SAS_k$

(d) ROC: source of the seeds

Fig. 7. Studying the effect of SNAP parameters on its performance on the CATH dataset. In (a) and (b) we plot the fractions of FP against the fractions of TP to show the effect of choosing different number of initial seeds on the performance of the algorithm. In (c) and (d) Comparison of the performance of SNAP on the CATH dataset for different values of $k$, used in calculating $SAS_k$.

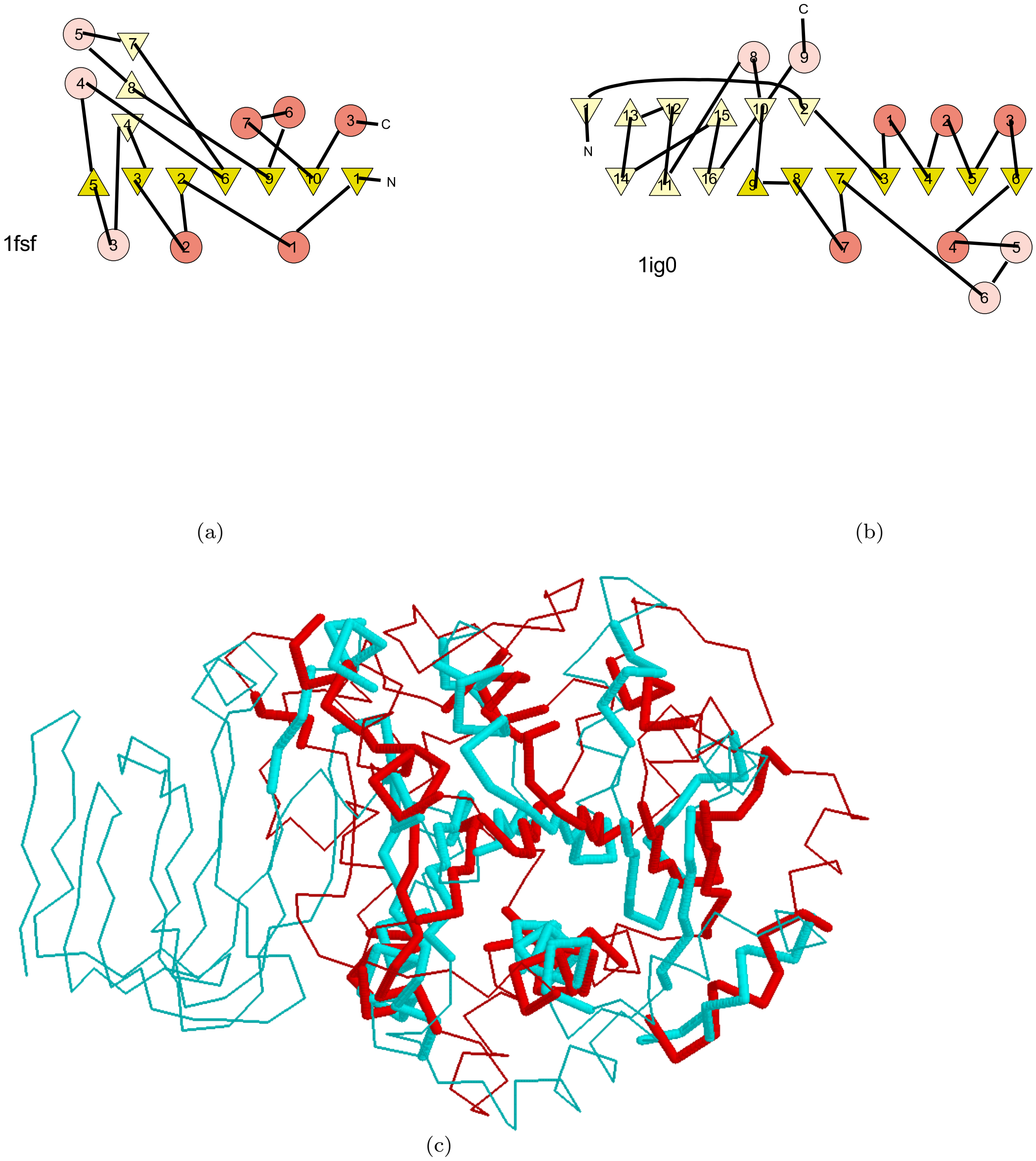


(a)

(b)

(c)

Fig. 8. A non-sequential alignment between (a) Leghemoglobin (2LH3:A, 153 residues) and (b) Cytochrome P450 BM-3 (2HPD:A, 471 residues). (c) SNAP alignment: Leghemoglobin in red and Cytochrome in blue. The $N_{mat}/rmsd$ scores were 118/3.37Å for SNAP, 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.



(a)

(b)

(c)

Fig. 9. TOPS cartoons [39] showing alignable secondary structure elements (alignable given the alignment in (c)) highlighted in red (helices) and yellow (strands) for a difficult non-sequential alignment of (a) Glucosamine-6-Phosphate Deaminase (1FSF:A, 266 residues) and (b) Thiamin Pyrophosphokinase (1IG0:A, 317 residues). Strands (triangles) and helices (circles) are numbered sequentially, independently. (c) SNAP alignment: 1FSF:A in red and 1IG0:A in cyan. The $N_{mat}/rmsd$ scores were 127/3.38Å for SNAP, 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.