The URMS-RMS hybrid algorithm for fast and sensitive local protein structure alignment

Golan Yona[1] and Klara Kedem[1,2]
[1] Department of Computer Science, Cornell University, Ithaca, NY
[2] Department of Computer Science, Ben Gurion University, Beer-Sheva, Israel

Abstract

We present an efficient and sensitive **hybrid algorithm** for local structure alignment of a pair of 3D protein structures. The hybrid algorithm employs both the URMS (Unit-vector Root Mean Squared) metric and the RMS metric. Initial transformations (rotations) are identified using the URMS algorithm. These rotations are then clustered and an RMS based dynamic programming algorithm is invoked to find the maximal local similarities for representative rotations of the clusters. Our algorithm searches efficiently the transformation space using a fast screening protocol. Given the transformation based parameters, the algorithm rigorously finds the optimal alignments. Statistical significance of the alignments is estimated using a model that accounts for both the score of the match and the RMS.

We tested our algorithm over the SCOP classification of protein domains. Our algorithm performs very well, its main advantages being (1) it combines the RMS and the URMS metrics (2) it searches extensively the transformation space (3) it can detect complex similarities and structural repeats (4) it is symmetric.

1 Introduction

Functional annotation of proteins as done today is based mostly on inference rather than on experiments. The pace in which new proteins are revealed rules out expensive in vivo experiments to define their functionality and therefore much effort is directed into improving methods for detecting similarity between proteins, as a basis for functional inference.

Sequence analysis is the most common method today for functional inference, and the analysis of new genes almost always starts with a simple database search. However, in many cases sequences diverged throughout evolution to the extent that their similarity cannot be detected with sequence comparison algorithms. Since protein structure is better conserved in evolution than sequence [1], detecting structural similarity can help identifying subtle similarities that are not detected by means of sequence comparison. Therefore, there is a strong interest in fast and sensitive structure comparison algorithms.

The importance of protein structure comparison pertains to other problems beyond homology detection.Advances in functional genomics, from protein structure prediction to studies of structure-function relationships, require a fast, automatic and well assessed method to compute the similarity of 3D protein structures. For example, it can be used to search for important structural and functional sites, such as active sites or interaction sites. Fold prediction systems require hundreds of thousands of structure comparisons per prediction, and both the speed and accuracy of the algorithm are important factors in the success of these systems. An effective structure comparison algorithm can be also instrumental in studies of the self-organization of the protein space [2, 3] as it can help in automatic domain detection and classification, and comparison of protein families.

1.1 Related work

The importance of protein structure comparison has been acknowledged and investigated by many over the years, starting with the pioneering work of Sander and coauthors [4, 5, 6] and followed by others. Algorithms that are based on matching distance matrices were introduced in [4],[5],[7] and [8]. Rackovsky and Scheraga [9] develop a measure based on local representation of protein chains by discrete analogues of curvature and torsion. A similar approach can also be found in [10]. Fischer et al [11] and Panac et al [12] apply geometric hashing to find matching pairs of α-carbon atoms between two molecules.

A number of groups extended classical sequence alignment ideas to handle structural alignment. Many of these methods employ dynamic programming (DP) techniques for computing protein structure alignment. The issue of creating a residue-residue scoring matrix is solved by applying a two-level DP, the lower level DP serving as a scoring matrix. In some cases chemical information on residues, and secondary structure information is taken into account (see, e.g., [13, 14, 15, 16, 17]). The above methods run in time $O(n^4)$ [17]. Gerstein and Levitt [18, 19] use iterative DP to find the optimal global structural alignment of proteins. They also present statistical estimates similar to those of BLAST.

Shindyalov and Bourne [20] build a structural alignment (CE) which starts with aligning fragment pairs (AFPs) and extending alignment paths till getting a single optimal alignment. Ortiz et al [21] report a method (MAMMOTH) which combines the URMS and a variant of MaxSub [22] as a basis for protein model comparison. This method is benchmarked on SCOP and CASP3. They use the URMS to compare all heptapeptide pairs of the two structures and transform these values to similarity scores; they further find the best local alignment by applying MaxSub-like [22] scheme of finding a transformation and looking for 3D neighborhood of residues under this transformation.

1.2 Our approach

Although several algorithms for protein structure comparison were proposed over the years, the problem is by no means solved. Specifically, in all the methods described above, the standard measure that evaluates the resemblance of two proteins structures is the root-mean-square (RMS) of aligned α-carbon 3D positions[1]. However, this measure can be overly sensitive to outliers and is highly length-dependent.

Chew et al [23] presented a new measure, *the unit vector RMS* (URMS). This method is based on a variant of the RMS distance, measuring differences between the global orientation vectors (unit vectors) at corresponding α-carbons of two proteins, rather than the differences between the positions of the α-carbons themselves. It is argued in that paper that this measure overcomes the disadvantages of the RMS. These advantages have been acknowledged in Kedem et al [24] where the measure has been used to locate a folding site, in [21, 22] where it is used to evaluate ab-initio predictions, and in [25] where it is applied to find the structural consensus and alignment of a protein family. The URMS metric is described in more detail in the next section.

In this paper we present a new and fast method to compute *pairwise local structural alignments*. Our method addresses the limitations of both the RMS metric and the URMS metric, while utilizing the merits of each. We study the notion of a structural transformation, and investigate the statistical properties of structural matches to derive sensible alignments and similarity measures.

Our algorithm searches the space of all possible alignments, under the set of allowed transformations, without having to exhaustively search this space or resorting to heuristic algorithms, such as sampling or match-length manipulation. To find the structural analogs of a protein structure we first use the URMS metric to detect all viable transformations. The RMS metric is then used to optimize the local match based on each

[1] In some cases the RMSD (root-mean-square-deviation) is used.

one of the top ranking transformations.

The hybrid algorithm we present here differs from previous algorithms in several ways: (1) it combines the advantages of the URMS and the RMS metrics; (2) it searches methodically the transformation space; (3) it can detect complex similarities, such as between multi-domain proteins; (4) it can detect structural repetition (multimers); (5) it is symmetric.

The paper is organized as follows: in Section 2 we review the URMS metric and present the main elements of our new algorithm. The scoring function is described in section 3. In section 4 we discuss the statistical properties of structural matches. Performance evaluation is reported in section 5.

2 Methods

2.1 Notations and basic definitions

We denote complete protein structures by uppercase letters $\mathbf{A}, \mathbf{B}$. Each protein structure is given with its complete set of x, y, z coordinates for all its atoms. We reduce this representation to the α-carbon backbone atoms $\mathbf{A} = \{\vec{A_i}\}_{i=1}^{n} = \{(Ax_i, Ay_i, Az_i)\}_{i=1}^{n}$ where n is the number of amino acids. We denote by **a** and **b** fragments of **A** and **B**, respectively, where a fragment is a short consecutive substructure of the protein.

2.1.1 The URMS distance

The URMS distance [23] is a variant of the RMS distance: instead of comparing residue positions we compare unit vectors. The compared vectors are the vectors between adjacent α-carbons along the protein backbone. We refer to these direction vectors as *unit* vectors since, for proteins, these vectors are all approximately of the same length (about 3.8Å). By chaining the unit vectors head-to-tail, we obtain the standard model of a protein as a sequence of α-carbons in space. Alternatively, we can place all of the unit vectors at the origin; the protein backbone is thus mapped into unit vectors in the unit sphere (see Figure 1).

Formally, given a protein structure **A**, its **unit vector model** is the set

$$\mathbf{U_A} = \{\vec{u_i}^A\}_{i=1}^{n-1}$$

where $\vec{u_i}^A = (\vec{A_{i+1}} - \vec{A_i})/||\vec{A_{i+1}} - \vec{A_i}||$.

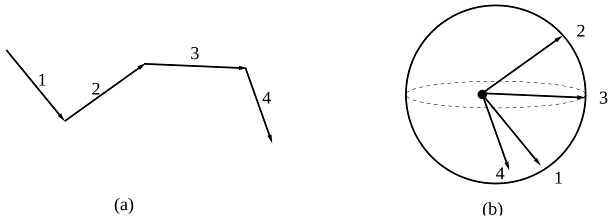

Figure 1: (a) A sketch of the protein backbone; the unit vector i connects amino acid i to amino acid $i+1$, ($i = 1, 2, 3, 4$) (b) the unit vectors translated to the origin (into the unit sphere).

The URMS distance between two protein structures **A** and **B** is defined as the minimal total Euclidean distance between their unit vector models, under rotation. Formally,

$$DIST_{urms}(\mathbf{A}, \mathbf{B}) = \min_R \{DIST_{euclid}(\mathbf{U_A}, \mathbf{U_B^R})\}$$

where $\mathbf{U_B^R}$ is the rotated unit vector model of **B**. The rotation R that minimizes the sum of Euclidean distances between the rotated unit vector models is referred to as the **URMS rotation**. Since the unit vector models of both proteins are centered at the origin there is no need to introduce translation.

Note that, just like the RMS distance, the URMS distance computation provides us with: (1) the minimal distance between the two proteins and (2) the rotation that minimizes that distance. However, unlike the RMS distance, no single outlier can have a large effect on the URMS distance, as the maximum squared distance between any two unit vectors is 4.

2.1.2 Structural agreement

Given two protein fragments **a** and **b** of length ℓ we say that the fragments are **in structural agreement under rotation R** if

$$DIST_{urms}(\mathbf{a}, \mathbf{b}) < T_{urms}$$

where T_{urms} is a preset threshold and R is the URMS rotation.

The fragment length ℓ is determined through optimization to be 8. This is consistent with the average length of a typical secondary structure element. To determine T_{urms} we study the distribution of distances between fragment pairs. As is shown in Figure 2 the distribution indicates the existence of two populations, of similar fragments and of non-similar fragments, suggesting a natural threshold of 0.6.

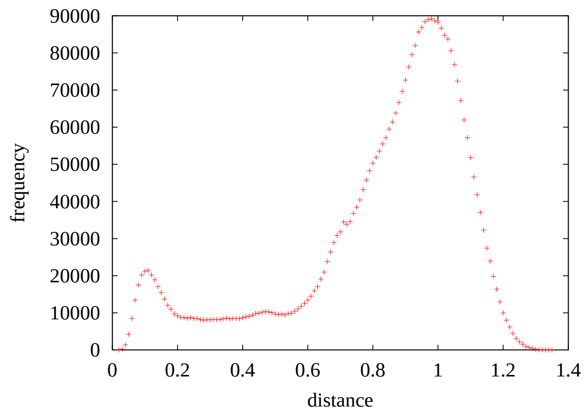

Figure 2: **Distribution of URMS distances**

2.2 Algorithm outline

Given a pair of $3D$ protein structures, our goal is to find the most similar substructures. Our algorithm works as follows

1. **Searching the rotation space:** Search for the complete set of similar fragment pairs using the URMS metric. For each pair of fragments determine the optimal rotation.
2. **Vector quantization:** Cluster rotations into clusters of similar rotations.
3. **Searching the reduced translation space:** For each cluster of similar rotations, identify the most consistent translation (as defined below) amongst its members. The cluster centroid and the most consistent translation define a candidate transformation.
4. **Alignment:** Given a candidate transformation (rotation and translation), find the best structural alignment using dynamic programming with an RMS-based scoring matrix.
5. **Output:** Report the highest scoring transformations and alignments.

A detailed description of each step if given next.

2.2.1 Searching the rotation space

If there is a single rigid transformation under which substructure **A** is very similar to substructure **B**, then one would expect to find multiple fragment pairs of **A** and **B** that are in structural agreement under this transformation. We explore the rotation space spanned by transformations that bring fragments of **A** and **B** to agreement (the set of **viable rotations**). Specifically, every pair of ℓ-length fragments $(\mathbf{a}, \mathbf{b})$ of **A** and **B**, is compared using the URMS metric. If a fragment pair is in structural agreement under rotation R, then R is considered a **viable rotation** and is added to the set.

Enhancements: The computation of $DIST_{urms}(\mathbf{a}, \mathbf{b})$ requires finding the optimal rotation for that fragment pair. To speed up the computation we approximate this distance with the **surface distance** $DIST_{surface}$ based on the length of edges connecting the unit vectors (see Figure 3). The optimal rotation is computed only for fragment pairs that are considered close according to the surface distance. The surface distance between **a** and **b** is defined as

$$DIST_{surface}(\mathbf{a}, \mathbf{b}) = \sum_{i=2}^{l} \left|\, ||\vec{u_i}^a - \vec{u_{i-1}}^a||^2 - ||\vec{u_i}^b - \vec{u_{i-1}}^b||^2 \,\right|$$

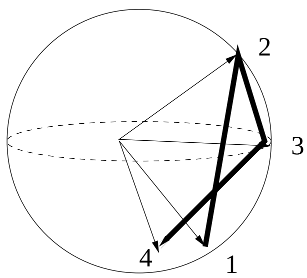

Figure 3: The unit sphere with four unit vectors; their surface distance is measured along the bold lines.

A fragment pair is said to be a **candidate fragment pair** if $DIST_{surface}(\mathbf{a}, \mathbf{b}) < T_{surface}$ where $T_{surface}$ is set to 2 (based on the distribution of surface distances, not shown). The intuition is that if the surface distance is small, then it is quite likely that the two sequences of unit vectors match well under the optimal rotation, and the URMS -based distance satisfies $DIST_{urms}(\mathbf{a}, \mathbf{b}) < T_{urms}$. Indeed, the correlation between these two measures is strong. On average, for 98% of all fragment pairs of URMS distance $< T_{urms}$, the surface distance is less than 4. This threshold, however, is too permissive as most fragments within surface distance of 4 are not in structural agreement as defined in section 2.1.2. To improve efficiency we set the surface distance threshold to $T_{surface} = 2$. With this threshold about 57% of the candidate fragment pairs are in structural agreement, at a moderate cost in sensitivity (eliminating 21% of the fragment pairs that are in structural agreement).

The main advantage of the surface distance is that it is much faster to compute than $DIST_{urms}$, as it does not involve computing the optimal rotation, thus allowing for fast rejection of many fragment pairs at this initial stage.

2.2.2 Vector quantization

The space of all viable rotations can be quite large. For two proteins of length 100 the number of fragment pairs that are in structural agreement can be on the order of hundreds or even thousands, each pair with its own optimal rotation. Our goal is to find the rotations that would generate good alignments. We assume that if the two structures are indeed similar most of the locally optimal rotations will tend to aggregate into one or a few clusters of similar rotations. However, in practice the situation is more complex. Slight differences between the locally optimal rotations are very common. Moreover, when the exact transformation between two structures cannot be formulated by means of translation and rotation alone but requires deformation as well (which is quite often the case) the locally optimal rotations might give rise to elongated and relatively wide clusters. Together with outliers these lead to a sparsely populated yet fairly well connected rotation space (see Figure 4). Clustering such spaces is a challenging task, especially when computational speed is a major issue.

We tested four different clustering algorithms for vector quantization, focusing on accuracy and speed. Other, more sophisticated clustering algorithms can clearly be employed as well, but at the cost of increased computation time. The four algorithms that were tested are: greedy algorithm, grid-like algorithm, connected components, and agglomerative pairwise clustering (details are omitted due to space constraints). Preliminary results suggest that the differences between the clustering algorithms do not substantially affect the performance (although at the time of writing we did not have the chance to test that exhaustively). Because of its speed and the symmetry (see below) we chose to proceed with the connected components algorithm.

Clustering rotation matrices vs. clustering angles: To cluster the rotations one needs a distance function between rotations. The **Frobenius distance** $DIST_F$ between rotation matrices is defined as the Euclidean distance between their representations as 9 dimensional vectors. This metric is less than optimal as the high dimensional space of rotation matrices seems to be sparsely populated even for closely related structures (Figure 4). Therefore, it is hard to determine typical within-cluster distances.

Motivated by these issues, as well as by computational considerations, we explored the compact angles representation for the rotation space. When clustering rotations using that representation, we save time both when computing distances between rotations and when selecting representative rotations (as described below) since angles can be averaged. It should be noted that in mapping rotation matrices to angles we detect a relatively small number of rotation matrices with a negative determinant (count for about 10% of the rotation matrices). These matrices represent isometry (rotation, translation and reflection) and are eliminated from the analysis for computational simplicity.

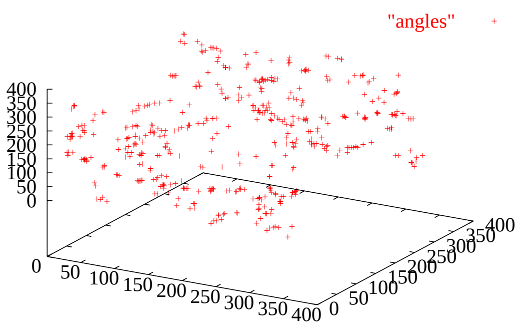

Figure 4: **The rotation space (angles representation).** All viable rotations for domain families 1.3.1.1 and 1.3.1.2 are plotted.

Symmetry and determinism. One of the major issues with structure comparison algorithms is their non-deterministic nature. Since the search space is too large to search exhaustively, all algorithms employ some kind of a heuristic to find good alignments, through sampling or directed search that can be highly influenced by the initial configuration (e.g. the initial orientation that is used to align the two structures). One of the consequences of this is that when comparing structures A and B the results might differ from the results one would get by comparing B with A. This sensitivity to the order in which the structures are being presented is clearly an artifact of the algorithm. Obviously, one can simply run the algorithm twice, once for A,B and the other for B,A, but at the cost of doubling the computation time. If there is an element of randomness, the symmetry is not guaranteed even then.

Our algorithm addresses this problem with only little increase of computational speed (or with no increase at all, depending on the clustering algorithm). If the clustering algorithm used is the connected components algorithm, then by definition this algorithm will find exactly the same rotations, inverted, in the inverse space when comparing B with A. This simply follows from the fact that $DIST_F(R_1, R_2) = DIST_F(R_1^T, R_2^T)$ and that $R^{-1} = R^T$ for rotation matrices. For all other clustering methods, our solution is to store both the viable rotation and the inverse rotation for each pair of consistent fragments. and treat them the same way we treat the original rotations.

Picking representative rotations A representative rotation is selected for each cluster that has a least n_{min} rotations (n_{min} is set to 5 in our tests). When the angles representation for rotations is employed, the cluster centroid (mean) is selected as the representative. When the matrix representation is employed, the centroid cannot be used since averaging rotation matrices will most unlikely generate a valid rotation matrix. Instead, we select as representative rotation the one that minimizes the total distance from all other rotation matrices in the cluster.

2.2.3 Searching the reduced translation space

While the URMS metric is very effective in finding the optimal rotation, it is not as effective for fine-tuned alignments of protein structures. In particular, different fragment pairs can have similar rotations but the translation might be different (see Figure 5), information that is missed when only relative positions are considered.

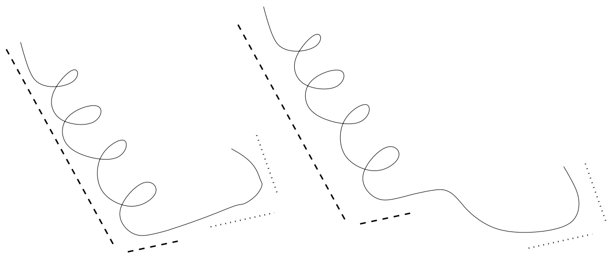

Figure 5: Both the dashed and the dotted lines show resemblance of fragments under rotation only, but for the dotted match the translation is different that that for the dashed lines.

One of the main novel elements of our algorithm is the combination of the URMS metric with the RMS metric. While the URMS metric is used to find the set of viable rotations, we use the RMS metric to align the structures. This is done in three steps.

The first step is to determine the proper translation for each cluster of rotations. Irrelevant for the URMS metric, finding the right translation is crucial for proper orientation of one structure with respect to the other, and for accurate alignment, when the absolute coordinates are considered.

Each cluster of rotations contains similar rotation matrices, derived from different pairs of fragments that are in structural agreement. Our assumption is that for the most part the cluster corresponds to fragments that can form a locally consistent alignment, therefore the proper translation should be one of the translations that are associated with the fragment pairs (**reduced translation space**). To find the best translation we run a search in the reduced translation space. The protein structures are first rotated according to the cluster rotation. For each fragment pair the structures are then translated such that the fragments are perfectly aligned to each other. Given this new orientation we evaluate the quality of the match by summing the total distance between the atoms in all fragment pairs. The final cluster translation is selected as the one that minimizes that total distance.

2.2.4 Alignment

Given the pair of rotation and translation, the structures are oriented accordingly. Our goal now is to find the most similar substructures. This is essentially a search for local similarity, and as such it requires a scoring function that would penalize far apart residues and gratify close residues. We study the statistical properties of structural alignments to derive an effective scoring function based on the RMS metric. This process is described in detail in section 3. Given the scoring function, we employ the dynamic programming algorithm to maximize the structural similarity. This is repeated for each cluster, given its representative rotation and translation, yielding the best alignment for this cluster.

2.2.5 Output

At this final stage of the algorithm we sift through the set of the alignments computed above. Each alignment that is deemed significant is reported. Due to repetitions or matches between different domains for multi-domain protein there might be more than a single significant match. We sort the matches based on their significance as described in section 4. To eliminate overlapping matches, all matches that overlap with previously reported matches are excluded. Each match is reported with its score, the RMS and the statistical significance.

3 Choosing the optimal scoring function

Many similarities between protein structures are actually localized to a single domain or structural motif. Detecting these motifs is a hard problem and has been a major hurdle for structure comparison algorithms. The main reason is that unlike sequence comparison, where the notion of the most similar subsequence is well defined (given the scoring matrix), no such notion has been agreed upon for protein structures, partly because there is no natural measure of similarity for individual residues as is the case for protein sequences.

To define a scoring matrix for local structure alignment we study the statistical properties of a large population of structural alignments (Leung and Yona, unpublished results). Given the alignments that were generated with CE[2] [20] we derive the distributions of atomic distances between aligned residues and unaligned residues (Figure 6). As the graph indicates, the two populations are well separated and the transition occurs around a threshold distance of $5\mathring{A}$.

We generate the scoring matrix by simply converting the atomic distances to similarity scores. This is repeated for every candidate transforma-

[2] Although we use another algorithm to generate these distributions, we believe that these distributions reflect the true properties of structural alignments as they are derived from significantly similar structures.

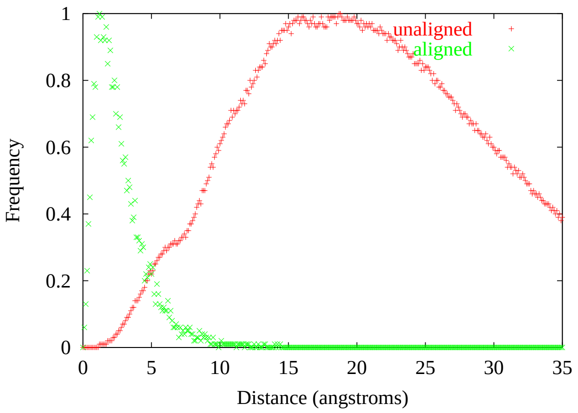

Figure 6: **Distribution of atomic distances between residues in structural alignments.**

tion (rotation and translation). Specifically, if the distance between residues A_i and B_j under the candidate transformation is d_{ij} then their similarity is defined as $s_{ij} = SHIFT - d_{ij}$ (a similar transformation was suggested in [5, 26].) Based on the graphs in Figure 6 we conclude that a reasonable range for the $SHIFT$ is between $4\mathring{A}$ and $7\mathring{A}$. Gap penalties are modeled using an affine function, because of its simplicity and speed of computation, and are set to a range similar to that of SHIFT, with the gap extension being roughly one order of magnitude less than the gap opening penalty. A range of values is explored for each parameter, and the parameters are optimized over a subset of the data set described in section 5, using the same methodology.

4 Statistical significance

The raw similarity score of the optimal alignment does not necessarily indicate true relationship, as it tends to increase just by chance with the length of compared structures. To assess the biological relevance of a match, one needs to know the expected score for a match between two random proteins of the same length. In our case, however, there are two different measures that can be used to score the match. The similarity score that is based on the atomic distances and accounts for gaps, and the RMS distance between aligned residues. It is the combination of both that will determine the relevance of a match. The higher the score and the lower the RMS, the better is the match.

To assess the significance we ask what is the probability to get a random match with the observed score and RMS. Formally, Given two proteins of length m and n and a match of length l with similarity score S and RMS R we compute the probability P

$$\begin{aligned} P(\mathbf{Match}/m,n) &= P(Score, RMS/m,n) \\ &= P(S/m,n)P(R/S,m,n) \end{aligned}$$

These probabilities can be derived from empirical background distributions for random proteins. However, Unlike sequences, random structures are not easy to generate. Moreover, a random structure is not likely be stable, as stereo-chemical constraints limit the conformation space to a small fraction of all possible shapes. A more sensible solution is to use existing but unrelated proteins. We generate this population from structure pairs that belong to different SCOP folds and classes [27]. The matches between these pairs are considered random, and a total set of 139047 alignments is generated and analyzed[3].

For the dynamic programming algorithm the structures compared can be viewed essentially as sequences with a position specific scoring function. Many of the principles that underlie the theory that was developed for random sequence matches [28, 29] hold here as well, as the properties required from the scoring function are easily met. Clearly there is at least a few positive matches, however, when considering all pairs of atoms the average score is negative, as only $O(n)$ positions can be aligned at the most (positive scores), while there are n^2 entries in the matrix. Under these conditions, the score of a random match between a sequence of length m and a sequence of length n has been shown to be centered around $a \cdot \ln(m \cdot n)$ where a is a function of the scoring matrix. Indeed, the distribution

[3]Even in that population of supposedly random pairs one might observe structural similarities due to recurrence of small motifs of secondary structures.

of similarity scores as a function of the product of the structures compared follows a logarithmic curve, as is shown in Figure 7a. The transformation of these score to zscores, by subtracting the expected average and dividing by the standard deviation results in a distribution that follows nicely the extreme value distribution [30] as is shown in Figure 7b.

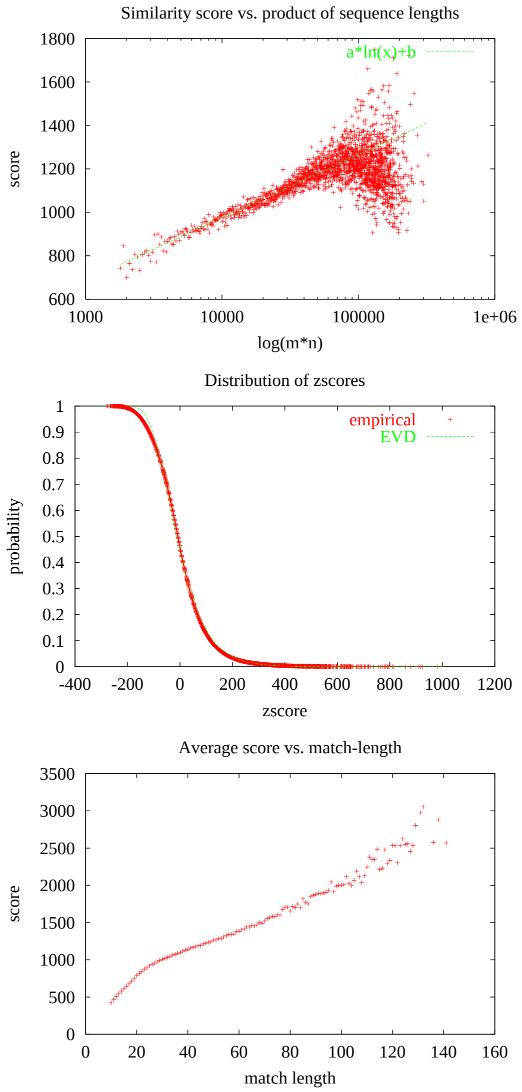

Figure 7: **Statistical properties of the scores of random matches.**

Since the score is linearly dependent on the length of the match (see Figure 7c) we can approximate $P(RMS/Score, m, n)$ by $P(RMS/l, m, n)$. That distribution is characterized in Figure 8a. Surprisingly, it seems that the RMS reaches saturation for match lengths around 100 residues, and the functional forms of the average as well as of the standard deviation (Figure 8b) are of tangent hyperbolic functions. The transformation to zscores results in a normal distribution, whose cumulative distribution is plotted in Figure 8c.

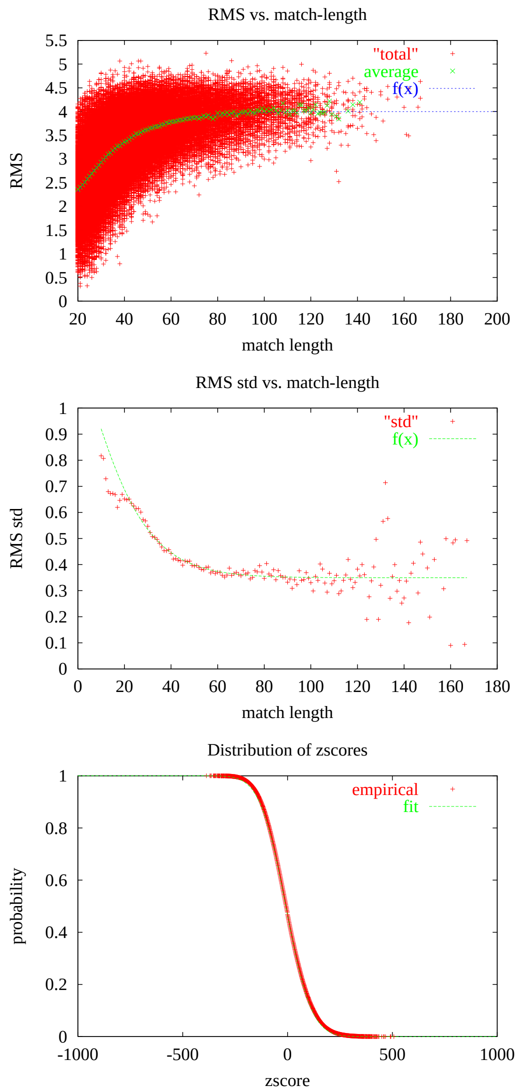

Figure 8: **Statistical properties of the RMS of random matches.**

Using the cumulative distributions for the zscores (Figure 7b,Figure 8b) we can estimate the probability to obtain an alignment with score $> S$ and RMS $< R$. The significance is estimated by the product of these two probabilities (the *pvalue*). In practice, however, the events are not independent, and the estimates might be overly optimistic.

5 Results

To assess our algorithm we tested it on the SCOP structural classification of proteins [27]. This hierarchy of protein domain families that is generated manually based on experts knowledge is an excellent benchmark for structure comparison algorithms. The SCOP hierarchy consists of class, fold, superfamily and family. Being able to recover this hierarchy with an automatic structure comparison algorithm is an indication of the algorithm's sensitivity and accuracy. Here we focused on the superfamily level. Our goal was to detect as many as possible family pairs that belong to the same superfamily, while minimizing the number of false positives.

5.1 Data sets

Our data set consists of 1287 domain families (SCOP 1.50). These families are organized into 814 superfamilies, 545 folds and 7 classes. A subset of 456 families (families within superfamilies that contain at least 3 families) is used in our analysis. For each family a structural representative is selected based on the minimal average sequence distance from all other members of the family and the SPACI index that measures the quality of the structure [31].

5.2 Global assessment

To evaluate the accuracy and sensitivity of the method we use the ROC1 index. ROC1 is simply the number of true positives detected before the first false positive[4]. To compute this index we compare the representative structure of each family with the 1287 other structures in our data set. The results are sorted based on the *pvalue*, and the number of true positives detected before the first false positive is counted. Finally, the results are summed over all queries.

Our definition of a false positive depends on what we consider to be a positive and what is deemed negative. We use two definitions of negatives. The first, more restrictive one considers all structures outside of the superfamily of the query structure to be negatives. However, structures within the same fold may be significantly similar as well. Our second definition of a negative is a more permissive one and considers all similarities within the same fold to be positives, while similarities outside the fold are considered negatives. The results in both cases are listed in table 1.

	Hybrid URMS -RMS	CE
Type of first false-positive:		
Different superfamily	270	179
Different fold	654	402

Table 1: **Performance evaluation results.** For each method, the number of true relationships that are detected before the first false connection occurs, is given. Results are given for the following types of false connections: different superfamily and different fold (see text for details).

We compare our results with those of CE [20]. This widely used algorithm is generously available from the CE website. As the results indicate, the hybrid algorithm performs very well. The running times of these algorithms are also comparable, with 1.5 seconds per alignment on average with our hybrid algorithm, compared to 5.75 seconds with CE.

6 Discussion

In this paper we present a new algorithm for protein structure comparison. Our algorithm collects information about significant rotations using the URMS metric, and generates multiple pairwise alignments using the RMS-based metric. As a result, the algorithm can report more than just a single significant alignment, such as between multi-domain proteins.

The algorithm has many computational options, catering for a large variety of applications,

[4] The popular ROC-N measure (e.g. ROC50) is inappropriate in this case because of the small sizes of the true positive sets. Other measures were used but are not reported due to space constraints.

and it can be tuned at different resolutions, depending on the desired speed, sensitivity and accuracy. Using the default set of parameters, our algorithm is very fast, with an average of 1.5 second per comparison. As we have shown, the algorithm compares very well with CE on the SCOP benchmark.

There are several future directions we intend to pursue. For example, inclusion of secondary structure information. It should be noted that the URMS metric captures the essence of a secondary structure (more than 85% of the fragments that are in structural agreement are also secondary structure consistent). First attempts to explicitly include secondary structure information in the scoring matrix did not improve performance, however, we are currently testing variations over this idea. We are also considering a more elaborate model for gap penalties based on the model described in [32], and refinement of the statistical model for significance assessment.

In this paper we adhered to the traditional set of rigid transformations (composed of rotations and translations). However, it is quite common to observe a transformation that involves deformation of the structure, that cannot be expressed only in terms of translation and rotation. Naturally, one might ask if different mathematical formulation is required to describe the relationships between functionally similar structures. Although this is outside the scope of this work, it has become clear to us that a new formulation has to be introduced. Our algorithm generates multiple pairwise alignments that correspond to different significant or popular rotations. We believe that this hybrid of rotations and translations can be useful to describe the exact transformation between the two structures when rigid transformations are inadequate.

References

[1] Holm, L. & Sander, C. (1996). Mapping the protein universe. *Science* **273**, 595-602.

[2] Yona, G., Linial, N. & Linial, M. (1999). ProtoMap: Automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins*, **37**, 360-378.

[3] Yona, G. & Levitt, M. (2000). Towards a complete map of the protein space based on a unified sequence and structure analysis of all known proteins. *In the proceedings of ISMB 2000*, 395-406, AAAI press, Menlo Park.

[4] G. Vriend and C. Sander, (1991). Detection of common three-dimensional substructures in proteins. *PROTEINS: Structure, Function and Genetics*, **11** 52-58.

[5] Holm, L. & Sander, C. (1993). Protein Structure Comparison by Alignment of Distance Matrices. *J. Mol. Biol.* **233**, 123-138.

[6] Holm, L. & Sander, C. (1994). Searching protein structure databases has come of age *Proteins: Structure, Function, and Genetics* **19**, 165-173.

[7] H.M. Griendley, P.J. Artymiuk, D.W. Rice & P. Willett, (1993). Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. Biol.* **229**, 707-721.

[8] Yee, D. & Dill, K. (1993) Families and the structural relatedness among globular proteins. *Protein Science* **2**, 884-899.

[9] Rackovsky, S. & Scheraga, H.A. (1978). Differential geometry and polymer conformations 2. Development of a conformational distance function. *Macromolecules* **13**, 1440-1453.

[10] Rackovsky, S. & Goldstein, D.A. (1998). Protein comparison and classification: A differential geometry approach. *Proc. Nat. Acad. Sci.* **85** 777-781.

[11] Fischer, D., Nussinov, R. & Wolfson, H. (1992). 3D substructure matching in protein molecules, *Proc. 3rd Intl. Symp. Combinatorial Pattern Matching*, Lecture Notes in Computer Science, Springer-Verlag. **644**, 136-150.

[12] Pennec, X. & Ayache, N. (1998). A geometric algorithm to find small but highly similar 3D substructures in proteins. *Bioinformatics* **14**, 516-522.

[13] Orengo, C.A., Brown, N.P. & Taylor, W.R. (1992). Fast structure alignment for protein databank searching. *PROTEINS: Structure, Function, and Genetics* **14**, 139-167.

[14] Taylor, W.R. & Orengo, C.A. (1989). Protein structure alignment. *J. Mol. Bio.* **208** 1-22.

[15] Taylor, W.R. & Orengo, C.A. (1990). A rapid method of protein structure alignment. *J. Theor. Bio.* **147** 517-551.

[16] Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. & Thornton, J. M. (1997). CATH-a hierarchic classification of protein domain structures. *Structure* **5**, 1093-1108.

[17] Taylor, W.R. (1999). Protein stucture alignment using iterated double dynamic programming. *Protein Science* **8**, 654-665.

[18] Gerstein, M. & Levitt, M. (1996). Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. *Proceedings of ISMB'96 Intelligent Systems for Molecular Biology*, AAAI Press, 59-66.

[19] Gerstein, M. & Levitt, M. (1998). Comprehensive assessment of automatic structural alignment against a manual standard, the SCOP classification of proteins. *Protein Sci.* **7**, 445-456.

[20] Shindyalov, I. N. & Bourne, P. E. (1998). Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* **11**, 739-747.

[21] Ortiz, A.R., Strauss, C.E.M. & Olmeda, O. (2002). MAMMOTH (Matching molecular models obtained from theory): An automated method for model comparison. *Protein Science* **11**, 2606-2621.

[22] Siew N, Elofsson A, Rychlewski L. & Fischer D. (2000). MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics* **16**, 776-85.

[23] Chew, L.P., Kedem, K., Huttenlocher, D.P., & Kleinberg, J. (1999). Fast detection of geometric substructure in proteins. *Journal of Computational Biology* **6(3-4)**, 313-325.

[24] Kedem, K., Chew, L.P. & Elber, R. (1999). Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins: Structure, Function and Genetics* **37**, 554-564.

[25] Chew, L.P. & Kedem, K. (2002). Finding the consensus shape for a protein family. *Proc. ACM 18th Symp. on Computational Geometry*, 64-73.

[26] Levitt, M & Gerstein, M. (1998). A Unified Statistical Framework for Sequence Comparison and Structure Comparison. *Proc. Natl. Acad. Sci. USA* **95**, 5913-5920.

[27] Hubbard, T. J., Ailey, B., Brenner, S. E., Murzin, A. G. & Chothia, C. (1999). SCOP: a Structural Classification of Proteins database. *Nucl. Acids Res.* **27**, 254-256.

[28] Karlin, S. & Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl Acad. Sci. USA* **87**, 2264-2268.

[29] Dembo, A., Karlin, S. & Zeitouni, O. (1994). Critical phenomena for sequence matching with scoring. *Ann. Prob.* **22**, 1993-2021.

[30] Gumbel, E. J. (1958). "Statistics of extremes". Columbia University Press, New York.

[31] Brenner, S. E., Koehl, P. & Levitt, M. (2000). The Astral Compendium for Protein Structure and Sequence Analysis. *Nucl. Acids Res.* **28**, 255-256.

[32] Gonnet, G. H., Cohen, M. A. & Benner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science* **256**, 1443-1445.