

PSI: Indexing Protein Structures for Fast Similarity Search

Orhan Camoglu, Tamer Kahveci and Ambuj K. Singh

Department of Computer Science University of California, Santa Barbara, CA 93106

ABSTRACT

Motivation: We consider the problem of finding similarities in protein structure databases. Current techniques sequentially compare the given query protein to all of the proteins in the database to find similarities. Therefore, the cost of similarity queries increases linearly as the volume of the protein databases increase. As the sizes of experimentally determined and theoretically estimated protein structure databases grow, there is a need for scalable searching techniques.

Results: Our techniques extract feature vectors on triplets of SSEs (Secondary Structure Elements). Later, these feature vectors are indexed using a multidimensional index structure. For a given query protein, this index structure is used to quickly prune away unpromising proteins in the database. The remaining proteins are then aligned using a popular alignment tool such as VAST. We also develop a novel statistical model to estimate the *goodness* of a match using the SSEs. Experimental results show that our techniques improve the pruning time of VAST 3 to 3.5 times while maintaining similar sensitivity.

Keywords: Protein structures, feature vectors, indexing

Contact: {orhan,tamer,ambuj}@cs.ucsb.edu

INTRODUCTION

The key problem in structural alignment of proteins is to find the optimal correspondence between the atoms in two molecular structures. It is not known which atoms of one structure correspond to the other. This makes an exhaustive search intractable and heuristics are frequently employed. The Root Mean Square Distance (RMSD) between the aligned atoms of two aligned structures is typically taken as a measure of the quality of the alignment. Given a correspondence, the problem of optimally aligning two structures through rotation and translation so that the RMSD is minimized can be solved efficiently in time linear in the number of atoms (Arun *et al.*, 1987).

There are three classes of algorithms for structural alignment of proteins (Eidhammer *et al.*, 2001). The first class performs structural alignment directly at the level of C_α atoms (Gerstein *et al.*, 1996; Holm *et al.*, 1993; Shindyalov *et al.*, 1998; Taylor *et al.*, 1989; Taylor

, 1999). The second class of algorithms first uses the SSEs (Secondary Structure Elements) to carry out an approximate alignment and then uses the C_α atoms. The final class of algorithms uses geometric hashing (Wolfson *et al.*, 1997; Holm *et al.*, 1995; Nussinov *et al.*, 1991).

Hierarchical algorithms are based on rapidly identifying correspondences between small *similar* SSE fragments of two proteins. The similarity of two fragments is defined using length and angle constraints. Fragment pairs that align well form the seed for extensive atom-level alignments. A significant speedup can be obtained since the number of SSEs is small and the 3-D structure within an SSE is constrained by hydrogen bonding. This is followed by a more detailed alignment of the atoms themselves. We discuss the VAST algorithm below. Other algorithms carrying out hierarchical alignment are (Alexandrov *et al.*, 1996; Koch *et al.*, 1996; Mizguchi *et al.*, 1995; Rufino *et al.*, 1994; Singh *et al.*, 1997).

The VAST algorithm (Madej *et al.*, 1995) carries out a hierarchical alignment beginning with SSEs. It begins with a bipartite graph: vertices on one side consist of pairs of SSEs from query protein and vertices on the other side consist of pairs of SSEs from target protein. An edge is inserted between two pairs of SSEs if they can be aligned well. A maximal clique is found in this bipartite graph; this defines the initial SSE alignment. This initial alignment is extended to C_α atoms by Gibbs sampling. A nice feature of the VAST program is its ability to report on the unexpectedness of the match through a *p-value*. This is computed by considering the size of the match, the size of the proteins, and the quality of the alignment.

In this paper, we consider the problem of finding similarities in protein structure datasets. Our techniques can be used to prune uninteresting proteins for a given query (or a set of queries) quickly. We propose to extract feature vectors corresponding to triplets of SSEs. Later, an R*-tree (Beckmann *et al.*, 1990) is built on this feature space using *Minimum Bounding Rectangles (MBRs)*. Our technique, called *PSI (Protein Structure Index)*, finds high quality seeds by aligning the SSEs that are similar to a given query protein. The proteins that do not have high quality seeds are pruned without further consideration. We also develop a novel statistical model to compute the *p*-

value of a seed. This value defines the goodness of this seed.

INDEXING PROTEIN STRUCTURES

Current techniques sequentially compare the given query protein to all of the proteins in the database to find similarities. Therefore, the cost of similarity queries increases linearly as the volume of the protein databases increase. We propose to reduce the protein structure search cost by pruning the database proteins that are not similar to a given query protein efficiently. We achieve this by building an index structure on the SSEs of the database proteins.

In order to construct the index structure, we approximate each SSE using a line segment in 3-D. For each SSE, we construct a set of SSE triplets by considering the SSEs in the local neighborhood around that SSE. For each triplet, we store information about pairwise distances and pairwise angles for all pairs of SSEs in that triplet. The pairwise distance information is a range of values obtained by considering a set of points around the center of the line segment approximation of each SSE. This range is defined by using the minimum and maximum of these distances between the set of points chosen from the two SSEs under consideration. The pairwise angle information is a single value that measures the angle between the line segment approximations of the two SSEs. Thus, we have a set of three range values and three angle values for each SSE triplet as the feature vector. This feature vector is an extent in six-dimensional space. These feature vectors are then indexed using an R*-tree (Beckmann *et al.*, 1990).

SEARCH TECHNIQUE

For a given query protein, our search technique (Camoglu *et al.*, 2003) runs in two phases:

- *Phase 1:* A set of feature vectors is obtained from the query protein, and the R*-tree is searched using an appropriate range with each of these vectors. Using the results of these range searches, a candidate set of database proteins is determined at the end of this phase.
- *Phase 2:* A pairwise structure alignment program, such as VAST, is then run on the candidate proteins to find the actual C_α alignments.

We elaborate on the first phase.

The range queries on the R*-tree find similar pairs of SSE triplets. Each such pair defines a mapping of three neighboring SSEs from the query protein to three neighboring SSEs in a database protein. A *score* is assigned to each mapping of SSEs of a triplet pair based on the inverse of the RMSD between the midpoints of the corresponding SSEs.

Once the alignments of triples of SSEs are determined, alignments of larger number of SSEs can be found by merging these results. We capture the correlation between mappings of triplet pairs by building a *Triplet Pair Graph (TPG)*. The vertices of TPG correspond to aligned triplet pairs. The weight of a vertex is defined as the score of the alignment of its corresponding triplet pair. An edge is placed between two vertices if they share two SSE mappings. Each connected component in this graph represents a mapping between the SSEs in the triplets of this connected component. We run a *Depth First Search (DFS)* algorithm on the TPG to find the *Largest Weight Connected Component (LWCC)*. The largest weight connected component of the TPG corresponds to the most similar subset of SSEs of database proteins and the query SSEs. We find an alignment of the SSEs by inspecting the subset. We start by constructing a bipartite graph on the LWCC. Unlike TPG, the bipartite graph consists of two disjoint vertex sets. The vertices in the first set correspond to the database protein's SSEs in the LWCC. The vertices in the second set correspond to the query protein's SSEs in the LWCC. The weight of an edge show how good the alignment is between the corresponding pair of vertices, and it is computed as the sum of the scores of the triplet pairs that maps these SSEs. We run a largest weight bipartite graph matching algorithm to find a mapping of the vertices in the two sets that maximizes the sum of edge weights. The resulting mapping defines a seed for each database protein.

Each seed represents an alignment of the query protein to a database protein in the feature space. Next, we use a novel statistical model and calculate the p-value of a seed. The p-value of a seed corresponds to the probability of having a seed at least as good as the given one in a randomly distributed space. Small p-values correspond to *unexpected* matches. We rank all the matches in the order of increasing p-values.

EXPERIMENTAL RESULTS

In order to evaluate the quality of our technique, we constructed a dataset D_{SDC} of single domain chains according to SCOP and VAST. From this dataset, we extracted 10 chains for those superfamilies that had at least 10 representatives. Call the resulting dataset D_{SF} . The dataset of query proteins, D_Q , was obtained by choosing a protein structure at random from each superfamily in D_Q . Dataset D_{SDC} consisted of 12,138 structures, dataset D_{SF} consisted of $180 * 10 = 1800$ structures, and dataset D_Q consisted of 180 structures.

Our first experiment set inspects the quality of the seeds found using the feature vectors. We classify the query protein (in D_Q) into one of the superfamilies using the k best results in feature space as follows. The logarithms of

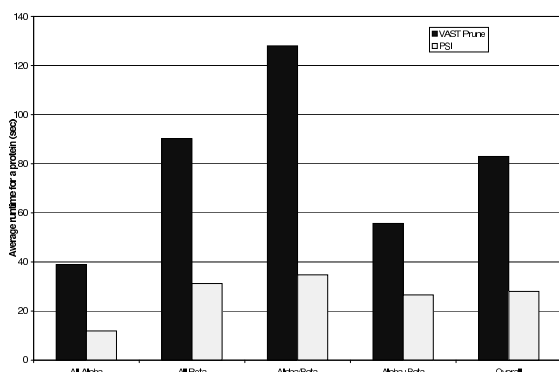


Fig. 1. Run time comparisons of VAST prune technique and PSI for various SCOP classes. Target set is the D_{SDC} dataset.

the p-values of the seeds of the results in each superfamily are accumulated. The query protein is classified as the superfamily that has the largest magnitude of this sum. In our experiments, more than 86% of the proteins are classified correctly using the first two nearest neighbors (NN). The quality increases to 88% for 3-NN, but the percentage drops for larger number of results. Even for 20-NN, more than 76% of the proteins are classified correctly.

We also tested our scheme to see how it performs as a pruning technique for an existing alignment tool such as VAST. For each protein in D_Q , we first ran VAST on D_{SF} , and computed how many proteins are returned in the answer set. We also ran PSI for the same protein on D_{SF} to obtain a candidate set. Later, we ran VAST on this set. We compared these two results to check whether PSI has pruned proteins that VAST considers relevant. According to our results, running VAST on the pruned dataset does not change the result set size significantly as opposed to running it on the whole dataset, and PSI has a recall of 98.2%.

We also compared the runtime performance of PSI with VAST's pruning step. VAST first finds seeds using SSEs of the query and protein. Then it computes p-values corresponding to these seeds. Finally the promising proteins (based on p-values) are considered for the expensive C_α alignment step. Since PSI aims to optimize the initial pruning, we considered the runtime of only the first two steps of VAST. For all proteins in D_Q , we ran PSI and VAST on D_{SDC} . Figure 1 shows a class-wise summary of the timing results. For all classes, PSI is significantly faster than VAST. The speedup is the highest for α/β proteins. The α/β proteins have more neighbors on the average. Because of that, VAST needs to inspect more seeds in these cases. However, PSI only considers the parts of proteins that are candidates for a similarity, and finds the seeds in linear time w.r.t. the number of SSEs in the proteins. More

results can be found in (Camoglu *et al.*, 2003).

ACKNOWLEDGEMENTS

This work was partially supported by NSF under grant BDI-0213903.

REFERENCES

- Alexandrov, N.N. and Fischer, D. (1996) Analysis of topological and nontopological structural similarities in the PDB: new examples from old structures. *Proteins*, **25**, 354–365.
- Arun, K.S., Huang, T.S. and Blostein, S.D. (1987) Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. on Pattern Anal. Machine Intel.*, **PAMI-9**, **5**, 698–700.
- Beckmann, N., Kriegel, H.-P., Schneider, R. and Seeger, B. (1990) The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. *SIGMOD*, 322–331.
- Camoglu, O., Kahveci, T. and Singh, A.K. (2003) PSI: Indexing protein structures for fast similarity search. *Technical Report, Department of Computer Science, University of California, Santa Barbara*, **2003-3**.
- Eidhammer, and Jonassen, B. (2001) Protein structure comparison and structure patterns – an algorithmic approach. *ISMB tutorial*.
- Holm, S. and Sander, C. (1993) Protein Structure Comparison by Alignment of Distance Matrices. *J. Mol. Biol.*, **233**, 123–138.
- Holm, S. and Sander, C. (1995) 3-D Lookup: Fast Protein Structure Database Searches at 90 % Reliability. *ISMB*, 179–187.
- Gerstein, M. and Levitt, M. (1996) Using iterative dynamic programming to obtain pairwise and multiple alignments of protein structures. *ISMB*, 59–66.
- Koch, I., Lengauer, T. and Wanke, E. (1996) An Algorithm for Finding Maximal Common Subtopologies in a Set of Protein Structures. *J. Comput. Biol.*, **3-2**, 289–306.
- Madej, T., Gibrat, J.F. and Bryant, S.H. (1995) Threading a database of protein cores. *Proteins*, **3-2**, 289–306.
- Mizuguchi, K. and Go, N. (1995) Comparison of spatial arrangements of secondary structural elements in proteins. *Protein Eng.*, **8**, 353–362.
- Nussinov, R. and Wolfson, H.J. (1991) Efficient Detection of Three Dimensional Structural Motifs in Biological Macromolecules by Computer Vision Techniques. *Proc. Natl. Acad. Sci. USA*, 10495–10499.
- Rufino, S.D. and Blundell, T.L. (1994) Structure-based identification and clustering of protein families and superfamilies. *J. Comput. Aided. Mol. Des.*, **233**, 123–138.
- Shindyalov, H.N. and Bourne, P.E. (1998) Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path. *Protein Eng.*, **11-9**, 739–747.
- Singh, A.P. and Brutlag, D.L. (1997) Hierarchical Protein Structure Superposition Using Both Secondary Structure and Atomic Representations. *ISMB*, 284–293.
- Taylor, W.R. (1999) Protein Structure Comparison Using Iterated Double Dynamic Programming. *Protein Sci.*, **8**, 654–665.
- Taylor, W.R. and Orengo, C.O. (1989) Protein structure alignment. *J. Mol. Biol.*, **208**, 1–22.
- Wolfson, H.J. and Rigoutsos, I. (1997) Geometric Hashing: An Introduction. *IEEE Comput. Sci. Eng.*, 10–21.