A Novel Method for Feature Identification of Proteins

Chandrasekhar Mamidipally[1,3], Santosh B. Noronha[1,4], Sumantra D. Roy[2]
[1] Department of Chemical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India. Ph: (91)2225767238, Fax: (91)2225723480. Email: chandra_m_sekhar@hotmail.com, noronha@che.iitb.ac.in
[2] Department of Electrical Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. Ph: (91)1126591084, Fax: (91)1126581606. Email: sumantra@ee.iitd.ac.in.
[3] Presenting author. [4] Corresponding author.

Submission to BIOCOMP'07

Keywords: Protein, Structure, Cluster, Feature

Abstract

With a rapidly growing database of protein structures, one needs fast algorithms for comparison of two protein structures, based on an efficient representation of a protein. As such, the problem has exponential time complexity, which is prohibitive if one has to perform the comparison at the residue level. One needs efficient representation methods to compare 3-D patterns of residues. In this paper, we propose the use of nearest-neighbour clustering as an efficient means to find out higher-level features in a large database of proteins. We also propose a method to estimate the optimal size of a biologically significant feature.

1 Introduction

Protein structure comparison is an important problem in bioinformatics [7, 32]. Structural similarity provides several insights into evolutionary and functional relationships between proteins; these insights are not usually evident from amino acid sequence alignment [11, 12, 16]. A protein may be represented as a collection of its constituent atoms - points in 3-D space, for example. In general, any method of structure comparison involves comparing a set of m points to another set of n points, in a k-dimensional space. As a first-level simplification, many approaches use the C_α atom of each amino acid to represent each residue in the protein [29]. As evident from structures deposited into the Protein Data Bank (hereafter, PDB) [3], proteins have up to about 600 residues. The number of entries in the PDB exceeds 25000 [24], and has been growing extremely fast presenting a further challenge for efficient feature identification.

Biswas and Chakraborty pose the problem as that of comparing two point sets, using the *Bottleneck Matching metric* [4], and propose an efficient approximation algorithm for the same. The nature of the general problem is however, combinatorial. Researchers often reduce the size of the problem by comparing significant features in the protein, as opposed to working at the residue level. Clearly, an algorithm with exponential time complexity for large proteins, in a rapidly expanding database is not desirable. An important parameter for comparison is the size of the significant features in a protein. One needs to search for features that are above a particular size threshold. Even if we ignore the problem of searching for an optimal threshold, an optimal solution to the structure comparison problem is possible only through exhaustive search (incurring an exponential time complexity). The aim of the above exercise is often to pull out motifs which remain conserved - forming the basis of classification into similar structural categories (this often indicates evolutionary relationships, or functional similarity). Examples include CATH [23], SCOP [18] and FSSP [13]. In a nutshell, one requires a protein structure representation, which facilitates structure comparison, and needs to compare as few features as possible.

In this paper, we propose the use of a fast method for detecting motifs in proteins (which may not necessarily correspond to secondary structures). We note that one would like to fit data to a line only if one has evidence that the data points are more-or-less distributed in a linear manner. Similarly, proteins often contain groups of highly clus-

tered atoms. Our clustering strategy pulls out such groups quickly. While an optimal solution requires exponential time complexity, our method obtains with a high probability, the required clusters of atoms in far less time as compared to that required for a exhaustive search. We propose a methodology to find the optimal threshold. To the best of our knowledge, no related work addresses all these issues.

The organization of the rest of the paper is as follows: in the remainder of this section, we briefly review motifs used in comparison techniques, and examine the sizes of the motifs used for the comparison. We then introduce our method for feature identification. We discuss cluster validation approaches and the effect of cluster size, and evaluate the utility of identified features for comparison of structures.

1.1 Motifs used in Structure Comparison

Protein structure comparison methods can be classified into three categories: methods based on the structure of common subsequences, methods that rely on specific secondary structure elements (hereafter, SSEs), and methods that look for generic 3-D patterns that are conserved. Other classification methods also exist *e.g.*, [7].

SSAP [21] and DALI [12] evaluate distance matrices toward identifying linear equivalence. Intra-atomic Euclidean distances of one protein are compared with to those on another protein, invariant to rigid body rotation and translation. DALI decomposes the distance matrices into incremental hexapeptide fragment-based submatrices to calculate initial correspondence, and then employs a Monte Carlo-based optimal alignment strategy. The approach of Rooman *et al.* [26] is based on distance between C_α atoms using the criteria of backbone dihedral angles. The clustering algorithm uses short polypeptide fragments of length containing 4 - 7 residues and uses root mean square deviation (RMSD) for comparison. The structural heterogeneity within each class is limited by fragments no longer than 7 residues. Another work addresses the backbone building problem [1]. The authors divide proteins into fragments of length 6 - 7 residues for common spatial arrangements of backbone fragments. The approach of [31] proposes the assembly of tertiary structures from fragments of length 5 - 9 residues with similar local sequences. Conklin [6] reviews several structural motif works (3-D objects described by the position of residues) and infers that all of these are in agreement with a fragment length of 6-8 residues. He also evaluates the performance of IMEM (Image MEMory) based on a large database of protein heptamer fragments. MAMMOTH [22] uses heptapeptides as features for comparison.

Levitt [17] predicts missing links in a protein using Segment Match Modeling where the optimal residue lengths of the main chain and the side chain targeted are 4 and 3 respectively. Fidelise *et al.* [8] predict loops in proteins using a fragment database for searching the target loops limited to length 4 residues only whereas van Vlijmen and Karplus [35] target loops up to 9 residue length for loop prediction. FAST [37] performs clique detection in *pair-graph* with each node represented by a pair of residues belonging to the query and target proteins, and an edge based on cut-off criteria imposed on intra-molecular distances between the two residue pairs i.e., nodes. Filtering is done by choosing overlapping pentapeptide fragments to produce initial best residue pairs forming nodes.

Shindyalov and Bourne [30] propose the idea of Combinatorial Extension (CE) to align two protein structures. The method uses an empirically determined fragment size of 8 to handle gaps and also increase the accuracy of comparison with an acceptable computational cost. Rossmann and Argos [28] discuss their method of comparison by rotating one protein about its center of mass with respect to the other and calculating the representative probability of the degree of structural parallelism. These probabilities are calculated for every third atom along the polypeptide chain implying a fragment size of residue length 3. Yang and Honig [36] use PrISM (Protein Informatics System for Modeling) to model protein sequences and structures. The structural similarity is based on geometric considerations alone, as against SCOP [18]. The method uses a critical parameter set empirically to 8 Å to compare distantly related structures.

SSE methods usually perform faster than sequential distance based approaches like DALI. However, they are more prone to SSE misalignments. KENOBI [34] compares proteins based on SSEs (containing at least four residue pairs) using a genetic algorithm. SARF [1] identifies SSEs by using prototypes of α helices and β sheets of fragment size 5 residues. The large ensemble of compatible pairs of SSEs are further compared using certain distance and angle constraints. A similar approach is used in VAST [10] where the SSE elements are compared using a graph theoretic approach. An edge is introduced between two SSE elements (nodes) if corresponding pairs of SSEs meet criteria for cut-off distance and conformational angles. The resulting correspondence graph was searched for maximal common subgraphs (cliques) to find the initial SSE alignment. Extension of alignment is then achieved using Gibbs Sampling technique. LOCK [33] represents SSEs as vectors and uses dynamic programming to obtain optimal superposition. SSM [15] is similar to LOCK, but is capable of handling mirror-symmetric structures.

The other approach to the problem of structure comparison is to find out similar groups of atoms in the two proteins independent of their location along the C_α backbone. A common example of such a method is to consider the protein as a graph with the atom being the vertices and the edges represented by the distance between the atoms within

the cutoff [19, 20] and finding the maximally connected subgraph (clique). The approach has exponential time complexity in the size of the graph [5]. Thus, the disadvantage of using a clique-finding algorithm is that there is no known algorithm which can solve the problem in polynomial time - in terms of either the number of vertices of the graph, or its number of edges [25]. An additional limitation of the above approach is the use of a threshold which governs the connectivity of one vertex to another: [29] for instance, consider cliques of diameter 12 Å. Local clustering of structural fragments has been implemented [27] to find recurrent patterns in proteins.

Another approach [2, 9] uses geometric hashing to match a 3D object (target protein structure) against one or more similar objects (query proteins). An object is represented by models where each model is constructed under different co-ordinate systems (formed by basis residues) called reference frames. In an offline preprocessing step, the reference frames are processed to form a highly redundant hash table. Subsequently, a lookup in a hash table is performed each time a model of query structure is searched for a match.

1.2 Clustering in Protein Feature Identification

We use the Nearest-Neighbour algorithm to identify groups of atoms as features, for use in 3-D structure comparison strategies. Jain and Dubes pose the Nearest-Neighbour Clustering problem as follows: "Given n patterns in d dimensional metric space, determine a partition of the patterns into K groups, or clusters, such that the patterns in a cluster are more similar to each other, than to patterns in different clusters" [14]. The authors show that the combinatorial nature of the problem makes exhaustive enumeration of all clusters clearly infeasible. They list out the use of heuristic methods - while optimality is not guaranteed, they work well in practice, in addition to being fast.

Two common strategies to alleviate the combinatorial problem are the $K-$ Means Clustering, and Nearest-Neighbour Clustering [14]. The former has a restricting requirement of having to specify K, beforehand. The advantage of the Nearest-Neighbour algorithm is that its computational time complexity is at worst *quadratic* in the number of patterns, and does not require *a priori* knowledge of K.

The advantage of our approach over others (reviewed in [7], for example) is that doing this for all possible configurations and then taking the optimal is equivalent to the optimal clique-based method. In other words, our method is scalable - an optimality-cost trade-off can be chosen, as required. In this paper, we show statistical evidence in favour of using this computationally efficient strategy, to give good results on protein structures. We also propose a valid estimate for the optimal cluster size - this forms a basis for the choice of a SSE-independent feature for protein structure comparison.

ALGORITHM RESIDUE_CLUSTERING

1. Randomly pick any C_α atom
2. Assign it to a new cluster
3. Randomly pick any C_α atom
4. Find its distance d_{min} to the centroid of the nearest cluster
5. IF $d_{min} \leq d_{thresh}$ THEN assign it to the corresp cluster; recompute the centroid ELSE assign it to a new cluster
6. IF no more C_α atoms THEN STOP ELSE GOTO Step 3

Figure 1. An overview of residue clustering.

2 METHODS

We considered 3500 protein structures from the Protein Data Bank (PDB). Four protein families (HIV Protease, Globins, Plastocyanins and Thioredoxins) were chosen based on their classification in the SCOP database. Proteins belonging to same class in SCOP have similar architecture and those having same folds have similar three-dimensional structure. We next chose proteins containing between 40 to 200 C_α atoms to prevent small peptides and multi-chain proteins from influencing our analysis. Each protein was then represented as a collection of C_α atoms in 3D-space. Two common strategies to alleviate the combinatorial natureof feature selection are K-Means Clustering, and Nearest-Neighbour Clustering [14]. In our context, the value of K is difficult to assess beforehand since the data set (i.e. the size of the protein) is not constant. We use the Nearest-Neighbour algorithm to identify groups of C_α atoms as features, for use in 3-D structure comparison strategies. A distance threshold d_{thresh} is used to identify neighbours. A summary of the Nearest Neighbourhood clustering procedure is shown in Figure 1.

Clustering is inherently ill-posed with a single instance of clustering unlikely to provide optimal partitioning; however when some order exists in the data, identification of reproducible clusters is feasible using a sampling approach. The general objective of the method is the estimation of the best set of centroids. This is a non-trivial problem because a single instance of clustering involves assignment of points to clusters (or creation of new clusters) a point at a time, with the points being read in in random order. The order in which points are randomly assigned has an influence on the final location of the centroids, and as a result, a sin-

gle clustering instance may not be representative (within a confidence interval) of the true centroid locations. This problem is resolved by repeatedly invoking the clustering algorithm, each time generating sets of centroids, and subsequently evaluating these centroids towards identification of conserved features. Therefore the algorithm proceeds in two phases: an iterative clustering phase, followed by combination of these iterations into a final clustering pattern.

Given a dataset of N points $L = \{C_{\alpha_1}, C_{\alpha_1}, \ldots, C_{\alpha_N}\}$, a clustering attempt according to the procedure in Figure 1 results in a partition P=$\{C_1, C_2, \ldots, C_K\}$ where C_j denotes a cluster belonging to partition P. If the clustering procedure were to reproducibly generate certain centroid sets over many iterations, then the probability of finding two C_α atoms i and j associated with such centroids would be high. An $N \times N$ binary matrix $TC(i,j)$ is used to denote the i^{th} and j^{th} atom co-membership in the same cluster ($TC_{i,j}$ =1 if i and j belong to the same cluster, 0 otherwise).

In the iterative clustering phase, the nearest-neighbour clustering algorithm is reemployed for M instances until a convergence criterion based on $TC(i,j)$ is achieved. For M clustering attempts, the resulting number of partitions will be $Q = \{P_1, P_2, \ldots, P_M\}$ for $j \in M$. The probabilities of the i^{th} and j^{th} C_α atoms occuring together (p_{ij}) in the M attempts of clustering is computed according to Equation 1.

$$p_{ij} = \frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{K_m} TC(i,j) \qquad (1)$$

The final averaging phase initially involves a ranking procedure on the $p(i,j)$'s obtained in the previous step, followed by subsequent agglomeration of i and j atoms according to their rank. The p_{ij}'s obtained are ranked in decreasing order. If p_{ij} and p_{kl} are tied, then the least distant pair is ranked higher. As a consequence of this ranking, an invariant order in which points may be clustered is now available. The same clustering procedure employed in the first phase is now employed on the ranked pairs, with the same threshold, to obtain the final partition $P = C_1, C_2, \ldots, C_K$, where K is the number of "meta"clusters and the centroids of these "meta" clusters will be referred to as "meta" centroids.

3 Results and Discussion

3.1 Efficacy of Nearest-Neighbour Clustering

Evaluation of all possible ordering of C_α atoms for identification of optimal features would involve exponential time complexity. In general, it is not possible to guarantee an iteration of the above algorithm to be optimal (or close to optimal), for an arbitrary data set. However, we note that proteins have numerous prominent groups of residues. For example, it may be assumed that the active sites of related proteins may be reasonably conserved in space, to ensure reactant accessibility, and subsequent reaction at the site. This domain-specific knowledge enables us to use the proposed method to advantage. The closer the residues in such a prominent group, the greater the chance of the algorithm identifying the entire group as a cluster - whatever be the order in which the atoms are selected. (Incidentally, this logic holds for relatively isolated C_α atoms as well.) This will however, not be true for any intermediate configuration. We note that two proteins are similar if they have similar configurations of prominent clusters of residues. In other words, prominent groups of C_α atoms alone (not necessarily corresponding to an SSE) are useful for 3-D structure comparison.

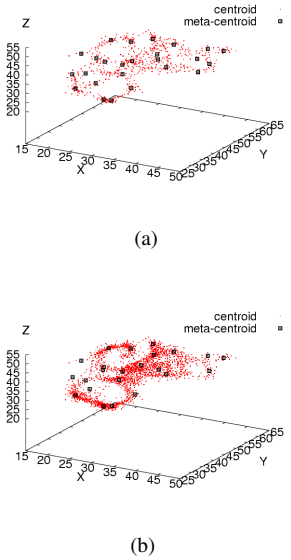
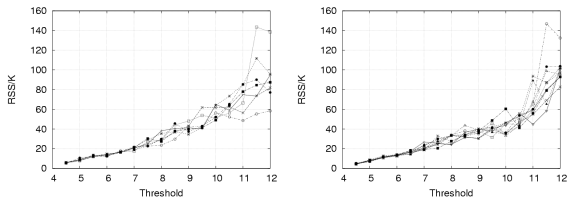


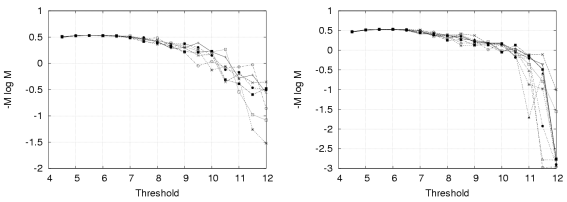
Figure 2. The impact of iterative clustering on the location of metacentroids for 1A2P. (a) 100 iterations, (b) 20000 iterations.

To verify the efficacy of our method in selecting prominent clusters very large number of iterations of the above clustering procedure was performed on various proteins. Figure 2 depicts the impact of iterative clustering on metacentroid identification for protein 1A2P. The centroid trace is almost similar for both 100 (Fig. 2(a)) and 20000 iterations (Fig. 2(b)). Prominent features are observed to be robust to different instances of clustering - metaclusters corresponding to prominent clusters are dense, and tightly packed, as opposed to others. For these experiments, we have used a empirically determined value of clustering threshold 6.0Å for each instance of clustering.



(a) (b)

Figure 3. RSS_K in proteins of (a) Thioredoxin family (b) Plastocyanin family.



(a) (b)

Figure 4. Measure M in proteins of (a) Thioredoxin family (b) Plastocyanin family

3.2 Feature size

The choice of cluster threshold d_{thresh} influences the clustering outcome. For a value of d_{thresh} less than the least inter-residue distance (distance between consecutive C_α atoms, ~ 3.8 Å), every residue forms a cluster in itself. On the other hand, if it equals or exceeds the maximum C_α-C_α distance, all residues will end up in one cluster. Therefore an optimal feature size involves a compromise between feature detail (large number of C_α atoms per cluster) and the number of clusters. The average residual sum of squares of C_α atoms from the centroids with which they are associated, on averaging over various proteins, serves to identify an optimal clustering threshold range.

$$RSS_K = \frac{1}{K} \sum_{m=1}^{K} \sum_{i=1}^{N_m} \|d(i,m)\| \tag{2}$$

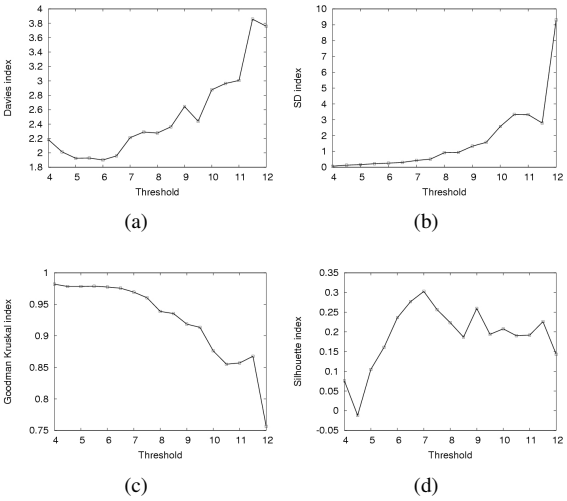
where $d(i,m)$ is the Euclidean distance between a C_α atom and its centroid, N_m represents the number of atoms in the m^{th} cluster, and K the number of clusters. The RSS_K versus threshold curves for proteins belonging to the thioredoxin and plastocyanin families are shown in Figure 3. Similar results are obtained for proteins belonging to α , β, and $\alpha + \beta$ classes of SCOP (data not shown).

To study the influence of threshold on clustering, we evaluated the degree of clustering measure M (Equation 3).

$$M = \frac{1}{K} \sum_{m=1}^{K} \sum_{i=1}^{N_m} \frac{1}{d(i,m)} \tag{3}$$

A smooth transition in M is seen as a function of threshold, for the thioredoxin and plastocyanin protein families in Fig. 4(a) and 4(b) respectively.

We evaluated several cluster validation indices to judge the compactness and separation of clusters, thereby identifying a desirable feature size. The following four indices were computed for the C_α clusters and the average curves for all the proteins tested were considered (a) Davies-Bouldin index, (b) SD validity index, (c) Goodman-Kruskal index, and (d) Silhouette index. Low values of Davies Bouldin (Fig. 5(a)) and Scattering Density (Fig. 5(b)) and high values of Goodman Kruskal (Fig. 5(c)) and Silhouette Index (Fig. 5(d)) indicate good clustering. We found overall that 6.0Å was an efficient cut-off threshold for clustering.



(a) (b)

(c) (d)

Figure 5. The effect of threshold on cluster validity. (a) Davies Bouldin Index (b) Scattering Density Index (c) Goodman Kruskal Index (d) Silhouette Index.

From inspection of the profiles of the various indices computed above, suitable features are of size 6-7.5 Å. The average number of C_α atoms per cluster (at 6 Å threshold) is 5 for α and $\alpha + \beta$ and 4 for β class proteins. The lower threshold (6 Å) results in more C_α atoms per cluster, but more clusters in total. The average backbone based peptide lengths for the proteins in our dataset were computed using end-to-end distances, and the results are presented in Table 1. It is evident from this table that the cluster diameter range we have obtained of 12-15 Å is consistent with

Backbone length	Mean	Std. Dev.	Largest
4	8.08	3.66	10.97
5	9.93	7.37	14.39
6	11.79	10.74	17.60
7	13.29	15.25	20.51
8	14.56	20.68	23.86

Table 1. The lengths (in Å) of backbone fragments of various sizes.

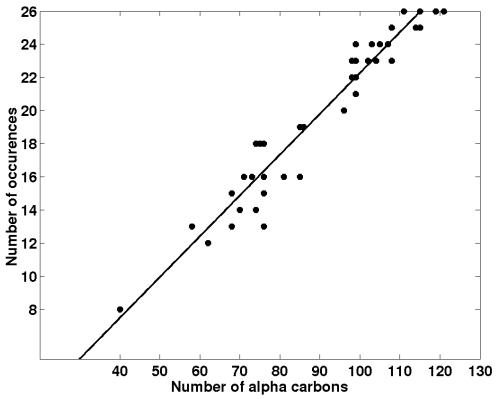


Figure 6. Protein size vs Number of clusters.

the (empirical) use of penta- and hexapeptides in existing backbone-based structure comparison methods. The range is also consistent with the empirical choice of 12 Å as a clique diameter for active site determination [29].

As the protein size increases, the number of features may be expected to increase, given the choice of a clustering distance threshold. An approximate linear fit between protein size and the number of clusters generated is evident from Figure 6, for all protein classes. With larger thresholds, larger features are obtained, but feature comparison may be expected to become more difficult, with identical features unlikely to occur.

3.3 Cluster significance

The clusters obtained in the Plastocyanin family were studied to find recurrence of substructures that underwent amino acid substitution during evolution (Table 2). Clusters with the same number of C_α atoms were considered. Each cluster is delimited by "/" and a cluster having non sequential residues has "-" as a separator. The amino acid labels in each clusters are ordered according to residue number. Groups of amino acids which are conserved across all the proteins in the family are highlighted using boxes (strings of amino acid labels).

The Cys-Ser-Pro-His (CSPH) motif occured in all the

No.	1AG6	1PLC	Matches	Score
1	PSGV	PSGV	4	56
2	CSPH	CSPH	4	70
3	NKG-VN	EKG-VN	4	52
4	QGAGM	QGAGM	5	78
5	I-TYK	I-EYS	1	27
6	G-SLA	G-SLA	3	48
7	DAAKI	DASKIS	4	63
8	GDDG-A	ADDG	3	36
9	GFP	AGFPH	3	37
10	SEED	MSEEDL	4	47
11	DEDE	DEDS	3	51
12	ASGE-TLT	PGE-ALS	2	59
13	KNN-GET	LL-KNN	3	33
14	LL-FLP	FVPS	2	37

Table 3. Correspondence of clusters of 1AG6 and 1PLC observed using dynamic programming. Underlined letters indicated unaligned residues.

Plastocyanins except 2PLT protein which had the amino acid serine (Ser) substituted by Glutamic acid (GLU) and 1KDI, which had a serine) substituted by threonine (Thr) as shown in boxes in Table 2. 1KDI and 1AG6 have 35.4% sequence similarity but posses high structural similarity. Similar results were obtained for other families: the thioredoxin family was discovered to have a CGPC (Cys-Gly-Pro-Cys) motif which describes its active site.

The cluster correspondence between 1AG6 and 1PLC was computed using dynamic programming-based string alignment using the Smith-Waterman algorithm and BLOSUM62, an amino acid substitution matrix is useful for finding distantly related sequences. Table 3 shows corresponding clusters of 1AG6 and 1PLC with a minimum length of size 3. We observed biologically significant clusters that superpose well due to retained substructure, despite substution with amino acids of similar properties.

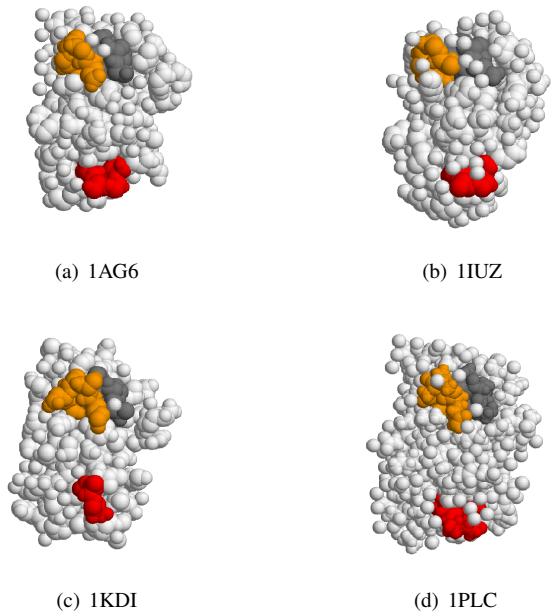
Figure 7 shows the three overlapping substructures (PSGV, CSPH, QGAGM) in all the proteins of Plastocyanin family. These three motifs are amongst the most common clusters across all plastocyanins. There are many other overlapping clusters; they differ in cluster size by 1 or 2. All the figures have been developed using RASMOL and have been rotated to show matched clusters. It is evident that structurally similar proteins have structurally similar substructures.

4 Conclusions

This paper presents an efficient and scalable algorithm to find generic 3-D features in a protein - the starting point for

PDB	Size	K	NS	Clusters (single letter codes for amino acids)
1AG6	99	22	13	PSGV/ CSPH/ DEDE/ QGAGM/ EKG-VN/ SEED/ GDDG-A/ G-SLA/ FSV-VT/ ASGE-TLT/ LL-FLP/ GD-GK/ GFP/ I-TYK/ DAAKI/ YKV/ VVF-SM/ VEV-EIVF/ HN-LL/ FY-V/ NAP/ KNN-GET
1BYP	99	24	12	AEV-IT/ PAGV/ DKKEV/ SSDGG/ AGAGM/ CAPH/ MPEED/ N-APGE/ AGFP/ LTEK/ SIA-VN/ K-VT/ DVTKIS/ GT-T/ H-LLN/ SGE/ LLG-N/ NDL-Y/ FK-EYS/ F-YKF/ DL-V/ LA/ VGK/ FVPS
1IUZ	98	23	9	CEPH/ PAGV/ AGFPH/ GDDG/ QIV/ AGAGM/ LNSK/ N-Y-D-DY/ TPG-VQ/ SLA/ EF-TVV/ VN-GE/ DADAIS/ FVPSK/ VYG-IT/ AI-RK/ IVF-VY/ KMT/ ISV/ KLG-N/ DEDAV/ AAGE-LS/ A
1KDI	102	23	11	GETG/ PAGA/ AKV-V/ CTPH/ DEND/ KSANM/ G-NFK/ SEDEP/ VSTP/ GTY/ DEVG/ EF-SFK/ SIT-TLT/ FYPD/ VSA-VK/ GEA-AK/ HN-LL/ EV-TLV/ FDI-TF/ Y-KG/ IV-SM/ SELKAA/ PGTVA
1OOW	99	23	9	PSGV/ CSPH/ DEDEI/ QGAGM/ GDDG-A/ SEED/ G-SEA/ VEV-IVF/ FSV-VT/ DAAKI/ FLPGD/ AS-LTEK-VN/ GEE/ GTY/ GFPH/ KVT/ VVF-SM/ KFY-VGK/ N-LL/ LL-KNN/ ETY/ NAP/ G
1PLC	99	24	9	IDV-F/ PSGV/ DEDS/ ADDG/ NKG-VN/ MSEEDL/ CSPH/ QGAGM/ AGFPH/ G-SLA/ AKGE/ SIS/ FVPS/ EF-VT/ LN/ PGE-ALS/ I-EYS/ VGK/ LL-KNN/ DASKIS/ KIV-EV/ NI/ VF-FY/ TF

Table 2. Clusters obtained at 6.0 Å threshold for Plastocyanin family. K is the number of clusters and NS the number of non-sequential clusters.



(a) 1AG6 (b) 1IUZ

(c) 1KDI (d) 1PLC

Figure 7. Highly conserved clusters in proteins of the plastocyanin family.

a protein comparison algorithm. We do not restrict - as other methods explicitly do - that our features be based on the protein backbone. The feature identification approach that we have employed is invariant to rigid body linear transformations. The scalability of our method is with respect to optimality of the solution, and the computational cost involved. Our method also computes an optimal threshold of 6 Å for a cluster, which we show to be consistent with the empirically determined values in successful 3-D structure comparison algorithms.

Thus, rather than randomly reading in points for a single attempt at clustering, the iterative approach enables us to rank sets of atoms in a defined order. Clearly, this order rewards points which co-cluster frequently and therefore, in greedy fashion, is biased towards identifying more conserved clusters. The advantage of our approach over others is that applying this algorithm for all possible configurations and then identifying a consensus, is equivalent to the optimal clique-based method. In other words, our method is scalable - an optimality-cost trade-off can be chosen, as required.

References

[1] N. N. Alexandrov, K. Takahashi, and N. Go. Common Spatial Arrangements of Backbone Fragments in Homologous and Non-Homologous Proteins. *Journal of Molecular Biology*, 225:5 – 9, 1992.

[2] O. Bachar, D. Fischer, R. Nussinov, and H. J. Wolfson. A computer vision based technique for sequence independent structural comparison of proteins. *Protein Engineering*, 6:279 – 288, 1993.

[3] F. C. Bernstein, T. F. Koetze, G. J. B. Williams, E. F. Meyer, M. D. Brice, J. R. Rogers, O. Kennard, T. Simanouchi, and M. Tasumi. The Protein Data Bank: A Computer-based

Archival File for Macromolecular Structures. *Journal of Molecular Biology*, 122:535 – 542, 1977.

[4] S. Biswas and S. Chakraborty. Fast Algorithms for Determining Protein Structure Similarity. In *Proc. Workshop on Bioinformatics and Computational Biology, 8th International Conference on High Performance Computing (HiPC)*, 2001.

[5] C. Bron and J. Kerbosch. Algorithm 457 - Finding All Cliques of an Undirected Graph. *Communications of the ACM*, 16:575 – 577, 1971.

[6] D. Conklin. Machine Discovery of Protein Motifs. *Machine Learning*, 21:125 – 150, 1995.

[7] I. Eidhammer, I. Jonassen, and W. R. Taylor. Structure Comparison and Structure Pattern. Technical Report Reports in Informatics (ISSN 0333-3590), Department of Informatics, University of Bergen, Norway, 1999.

[8] K. Fidelise, P. S. Sterne, D. Bacon, and J. Moult. Comparison of Systematic Search and Database Methods for Constructing Segments of Protein Structure. *Protein Engineering*, 7:953 – 960, 1994.

[9] D. Fischer, O. Bachar, R. Nussinov, and H. J. Wolfson. An efficient automated computer vision based technique for detection of three-dimensional structural motifs in proteins. *Journal of Biomolecular Structure and Dynamics*, 9:769 – 789, 1992.

[10] J. F. Gibrat, T. Madej, J. L. Spouge, and S. H. Bryant. The VAST Protein Structure Comparison Method. *Biophysics Journal*, 72:MP298, 1997.

[11] N. V. Grishin. Fold change in evolution of protein structures. *Journal of Structural Biology*, 134:167 – 185, 2001.

[12] L. Holm and C. Sander. Protein Structure Comparison by Alignment of Distance Matrices. *Journal of Molecular Biology*, 233:123 – 138, 1993.

[13] L. Holm and C. Sander. The FSSP Database: fold classification based on structure-structure alignment of proteins. *Nucleic Acid Research*, 24:206 – 209, 1996.

[14] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall Publication, New Jersey, 1988.

[15] E. Krissinel and K. Henrick. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica D*, 60:2256 – 2268, 2004.

[16] A. M. Lesk and C. Chothia. How different amino acid sequences determine similar protein structures: The structure and evolutionary dynamics of the globins. *Journal of Molecular Biology*, 136:225 – 230, 1980.

[17] M. Levitt. Accurate Modeling of Proteins Conformation by Automatic Segment Matching. *Journal of Molecular Biology*, 226:507 – 533, 1992.

[18] L. LoConte, B. Ailey, T. J. P. Hubbard, S. E. Brenner, A. G. Murzin, and C. Chothia. SCOP: A Structural Classification of Proteins Database. *Nucleic Acid Research*, 28:257 – 259, 2000.

[19] Y. Martin, M. Bures, E. Dahaner, J. DeLazzer, and I. Lico. A Fast New Approach to Pharmacophore Mapping and its Application to dopaminergic and benzodiazepine Agonists. *Journal of Computer Aided Molecular Design*, 7:83 – 102, 1993.

[20] E. M. Mitchell, P. J. Artymiuk, D. W. Rice, and P. Willet. Use of Techniques Derived from Graph Theory to Compare Secondary Structure Motifs in Proteins. *Journal of Molecular Biology*, 212:151 – 166, 1989.

[21] C. Orengo and W. R. Taylor. SSAP: Sequential Structure Alignment Program for Protein Structure Comparison. *Methods in Enzymology*, 266:617 – 635, 1996.

[22] A. R. Ortiz, C. E. M. Strauss, and O. Olmea. Mammoth (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science*, 11:2606 – 2621, 2002.

[23] F. M. G. Pearl, D. Lee, J. E. Bray, I. Sillitoe, A. E. Todd, A. P. Harrison, J. M. Thornton, and C. A. Orengo. Assigning Genomic Sequences to CATH. *Nucleic Acid Research*, 28:277 – 282, 2000.

[24] `http://www.rcsb.org/pdb/`, April 2004.

[25] E. M. Reingold, J. Neivergelt, and N. Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1977.

[26] M. J. Rooman, J. Rodriguez, and S. J. Wodak. Automatic Definition of Recurrent Local Structure Motifs in Proteins. *Journal of Molecular Biology*, 213:327 – 336, 1990.

[27] M. J. Rooman, J. Rodriguez, and S. J. Wodak. Automatic definition of recurrent local structure motifs in proteins. *Journal of Molecular Biology*, 213:327 – 336, 1990.

[28] M. G. Rossmann and P. Argos. Exploring Structural Homology of Proteins. *Journal of Molecular Biology*, 105:75 – 95, 1976.

[29] R. B. Russell. Detection of Protein Three-Dimensional Side-chain Patterns: New Examples of Convergent Evolution. *Journal of Molecular Biology*, 279:1211 – 1227, 1998.

[30] I. N. Shindyalov and P. E. Bourne. Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path. *Protein Engineering*, 11:739 – 747, 1998.

[31] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions. *Journal of Molecular Biology*, 268:209 – 225, 1997.

[32] A. P. Singh and D. L. Brutlag. Protein Structure Alignment: A Comparison of Methods. Technical report.

[33] A. P. Singh and D. L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. *International Conference on Intelligent Systems in Molecular Biology*, pages 284 – 293, 1997.

[34] J. D. Szustakowski and Z. Weng. Protein Structural Alignment Using Genetic Algorithm. *PROTEINS: Structure, Function, and Genetics*, 38:428 – 440, 2000.

[35] H. W. T. van Vlijmen and M. Karplus. PDB-based Protein Loop Prediction: Parameters for Selection and Methods for Optimization. *Journal of Molecular Biology*, 267:975 – 1001, 1997.

[36] A. Yang and B. Honig. An Integrated Approach to the Analysis and Modeling of Protein Sequences and Structures. I. Protein Structural Alignment and a Quantitative Measure for Protein Structural Distance. *Journal of Molecular Biology*, 301:665 – 678, 2000.

[37] J. Zhu and Z. Weng. Fast: A novel protein structure alignment algorithm. *Proteins: Structure, Function and Genetics*, 58:618 – 627, 2005.