Mining Protein Family Specific Residue Packing Patterns From Protein Structure Graphs

Jun Huan[1], Wei Wang[1], Deepak Bandyopadhyay[1], Jack Snoeyink[1], Jan Prins[1], Alexander Tropsha[2]
[1]Department of Computer Science, University of North Carolina at Chapel Hill
{huan, weiwang, debug, snoeyink, prins}@cs.unc.edu
[2]The Laboratory for Molecular Modeling, Division of Medicinal Chemistry and Natural Products, School of Pharmacy, University of North Carolina at Chapel Hill
tropsha@email.unc.edu

ABSTRACT

Finding recurring residue packing patterns, or spatial motifs, that characterize protein structural families is an important problem in bioinformatics. To this end, we apply a novel frequent subgraph mining algorithm to three graph representations of protein three-dimensional (3D) structure. In each protein graph, a vertex represents an amino acid. Vertex-residues are connected by edges using three approaches: first, based on simple distance threshold between contact residues; second using the Delaunay tessellation from computational geometry, and third using the recently developed almost-Delaunay tessellation approach.

Applying this approach to a set of graphs representing a protein family from the Structural Classification of Proteins (SCOP) database, we typically identify several hundred common subgraphs equivalent to common packing motifs found in the majority of proteins in the family. We also use the counts of motifs extracted from proteins in two different SCOP families as input variables in a binary classification experiment using Support Vector Machines. The resulting models are capable of predicting the protein family association with the accuracy exceeding 90 percent. Our results indicate that graphs based on both almost-Delaunay and Delaunay tessellations are more sparse than contact distance graph; yet the former afford similar accuracy of classification as the latter. The protein graph mining and classification approaches developed in this paper can be used for rapid and automated annotation of protein structures determined in structural genomics projects.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Science; H.2.8 [**Database Applications**]: Data Mining; I.3.5 [**Computational Geometry**]:

RECOMB'04, March 27–31, 2004, San Diego, California, USA.

General Terms

Algorithms

Keywords

Subgraph Mining, Delaunay Tessellation, almost-Delaunay, Protein Classification

1. INTRODUCTION

1.1 Spatial Motif Discovery in Proteins

Recurring substructural motifs in proteins reveal important information about protein structure and function. Common structural fragments of various sizes have fixed 3D arrangements of residues that may correspond to active sites or other functionally relevant features, such as Prosite patterns [12]. Identifying such *spatial motifs* in an automated and efficient way may have a great impact on protein classification [4], protein function prediction [9] and protein folding [20].

Several research groups have addressed the problem of finding spatial motifs by using computational geometry/computer vision approaches. A protein is typically represented as a set of points in R^3 and the problem of (pair-wise) spatial motif finding is often formalized as the Largest Common Point set (LCP) problem: identifying the largest common subset of two sets of points [27]. A number of variations to this problem have been explored, which include approximate LCP problem [4, 16] and LCP-α: identifying a subset of LCP that approximates LCP with a factor α characterizing the degree of approximation [8].

1.2 Application of Graph Theory to Molecular Structures

Chemical graphs have long been used to study molecules and macromolecules such as organic compounds, nucleic acids, and proteins. Recurring substructures in a group of compounds with similar biological activity can be identified by representing these compounds as undirected graphs, then finding frequent subgraphs. The recurring substructures can indicate chemical features responsible for compounds' activities [6, 3]. In another example, RNA secondary structure has been modeled by a labeled tree in which each node is a paired base and the parent/sibling relation is determined by the nesting of base pair bonds. The similarity between two RNA structures is measured by the largest approximately matched

subforests [11]. Many variations of this basic model exist, including rooted vs unrooted trees [19, 33], different granularity levels of representation [25], and different similarity measures [36].

Protein structures have been modeled using graph representations in many applications, including identification of active site clusters, folding clusters, aromatic clusters in relation to thermodynamic stability, and the analysis of protein-protein interaction. (See [31] for a recent and comprehensive review on applying graph theory to protein structure analysis.) The choice of a graph representation is the key in the analysis of protein structures. Several representations have been developed, ranging from coarse representations in which each node is a secondary segment [10] to fine representations in which each node is an atom [18].

We form graphs whose vertices represent the amino acids, using the coordinates of the C_α atoms and labeling by residue type. Two types of edge may connect residues: a bond edge that connects two residues that are adjacent in the primary sequence, or a proximity edge that connects two (non-bonded) residues identified as spatial neighbors by the following different criteria.

We consider three kinds of proximity edges. The first representation is a contact distance graph (CD) in which vertex-residues are connected with a proximity edge if the C_α atoms are found within a given distance δ of each other. The second representation is the Delaunay tessellation from computational geometry. This approach recognizes natural nearest neighbor residues by finding four-tuples that define tetrahedra with an empty sphere property [7]; this tessellation is dual to the Voronoi diagram. The Delaunay tessellation graph (DT) has been used to analyze protein packing [24, 29] and structure [22, 26, 34, 32, 28] The third representation, derived from the recently-defined almost-Delaunay edges [1], expands the set of Delaunay edges to account for perturbation or motion of point coordinates, controlled by a parameter ϵ. Thus, we actually define a family of graphs, $AD(\epsilon)$, that interpolates between the CD and DT representations so that $DT \subseteq AD(\epsilon) \subseteq CD$ for all $\epsilon \geq 0$. This representation allows us to reduce the number of edges in a CD graph without losing common patterns due to imprecise positioning of the protein's C_α atoms.

Several algorithms have been developed recently in the data mining community to find all frequent subgraphs of a group of general graphs [21, 35, 14, 13]. These algorithms can be roughly classified into two types: The first type uses a level-wise search scheme to enumerate the recurring subgraphs, such as AGM [17] and FSG [21]. The second type uses a depth-first enumeration for frequent subgraphs, which usually has better memory utilization and therefore better performance [35, 3, 14]. In fact, depth-first search can outperform FSG, the current state-of-the-art level-wise search scheme, by an order of magnitude [35].

Applying frequent subgraph mining to find common patterns for a group of proteins is a non-trivial task. As we reported earlier [14], the total number of frequent subgraphs for a set of graphs grows exponentially as the graph size increases. For instance, for a moderate protein dataset (about 100 proteins with the average of 200 residues per protein), the total number of frequent subgraphs can be extremely high ($\gg 1$ million). Because the underlying operation of subgraph isomorphism testing is NP-complete, it is critical to minimize the number of frequent subgraphs that need to be considered. This motivated us to investigate different graph representations for systematically simplified versions.

In our experimental study, we use frequent subgraph mining to identify subgraphs common to proteins of a given structural family found in the SCOP database [23]. The counts of subgraphs in different proteins are then used as input variables for a binary classification task to distinguish between two protein families in the SCOP database. The support vector machine approach is used to construct the classifier. We find that AD graphs afford a significant reduction in size of the graph representation, yet the features extracted from such graphs produce the highest classification accuracy. We suggest that frequent subgraph mining can be used to identify packing motifs that are highly specific to individual protein families providing opportunities for rapid and automated protein annotation.

The remainder of the paper is organized as follows. Section 2 presents definitions for the subgraph isomorphism and discusses various graph representations of proteins. Section 3 presents the data structure and the algorithm for subgraph mining and Section 4 presents the results of our study of several protein families from the SCOP database, including eucaryotic and procaryotic serine proteases and nucleotide binding proteins.

2. METHODOLOGY

2.1 Labeled Graph

We define a **labeled graph** G as a five element tuple $G = \{V, E, \Sigma_V, \Sigma_E, \delta\}$ where V is a set of vertices and $E \subseteq V \times V$ is a set of undirected edges. Σ_V and Σ_E are sets of vertex labels and edge labels respectively. The labeling function δ defines the mappings $V \rightarrow \Sigma_V$ and $E \rightarrow \Sigma_E$. Without loss of generality, we assume that there is a total order $\geq$ on each label set Σ_V and Σ_E.

A labeled graph $G = (V, E, \Sigma_V, \Sigma_E, \delta)$ is *isomorphic* to another graph $G' = (V', E', \Sigma'_V, \Sigma'_E, \delta')$ iff there is a bijection that preserves labels $f\colon V \rightarrow V'$ such that:

$$\begin{aligned} \forall u \in V, \delta(u) &= \delta'(f(u)) \\ \forall u, v \in V, \Big(((u,v) \in E &\Leftrightarrow (f(u), f(v)) \in E') \\ \wedge\, \delta(u,v) &= \delta'(f(u), f(v)) \Big) \end{aligned}$$

The bijection f defines as an *isomorphism* between G and G'. If G and G' refer to the same graph, f defines an *automorphism*.

A labeled graph $G = (V, E, \Sigma_V, \Sigma_E, \delta)$ is an *induced subgraph* of graph $G' = (V', E', \Sigma'_V, \Sigma'_E, \delta)$ iff $V \subseteq V'$, $E \subseteq E'$, $\forall u, v \in V, ((u,v) \in E' \Rightarrow (u,v) \in E)$, $\forall u \in V, (\delta(u) = \delta'(u))$ and $\forall (u,v) \in E, (\delta(u,v) = \delta'(u,v))$.

A labeled graph G is *induced subgraph isomorphic* to a labeled graph G', denoted by $G \subseteq G'$, iff there exists an induced subgraph G'' of G' such that G is isomorphic to G''. An example of labeled graphs and an induced subgraph isomorphism is presented in Figure 1. In the rest of this paper, the term "subgraph" will mean "induced subgraph" unless stated otherwise.

Given a set of graphs $\mathcal{G}$ (referred to as a *graph database*), the *support* of a graph G is defined as the fraction of graphs in $\mathcal{G}$ of which G is a subgraph. For a graph database $\mathcal{G}$, we choose a threshold $0 < \sigma \leq 1$, and say that G is *frequent* iff its *support* is at least σ. The problem of *Frequent Subgraph Mining* is to identify all frequent (connected) subgraphs of $\mathcal{G}$. Figure 3 shows all frequent subgraphs with $\sigma = 2/3$ in the three graphs of Figure 1.

2.2 Building Protein Graphs

Since the protein backbone trace defines the overall protein conformation, we choose the C_α atoms as the nodes of protein graphs. Based on this simplified protein model, we compute edges using three different approaches.

In the first representation, we connect two points by an edge if the distance between them does not exceed certain threshold δ. Since in principle we are interested in defining nearest neighbor residues located within the physical interaction distance, we chose

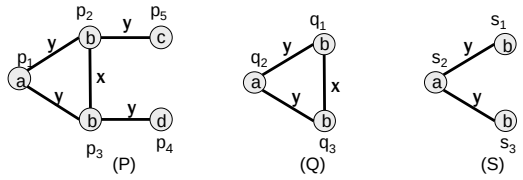

Figure 1: Example of a graph database $\mathcal{G}$ of three labeled graphs with an induced subgraph isomorphism. We assume that the node and edge labels are ordered s.t. $a > b > c > x > y$. The mapping $q_1 \rightarrow p_2$, $q_2 \rightarrow p_1$, $q_3 \rightarrow p_3$ represents an induced subgraph isomorphism from graph Q to P. Throughout this paper, we use the order $a > b > c > d > x > y > 0$.

$\delta = 10$ Å as the threshold. As stated in the Introduction, we refer to this distance-threshold dependent graph as the CD graph.

In the second representation, all nearest neighbor residues connected by edges are defined using Delaunay tessellation. This tessellation [7] is defined for a finite set of points by an *empty sphere property*: A pair of points is joined by an edge iff one can find an empty sphere whose boundary contains those two points. The Delaunay captures neighbor relationships in the sense that there is a point in space that has the chosen two points as closest neighbors. The Delaunay is dual to the Voronoi diagram—two points are joined by an edge in the Delaunay iff their Voronoi cells share a common face. Figure 2 illustrates the Delaunay in 2D with solid lines, and the dual Voronoi with dashed). We used the Quickhull [2] program to compute the Delaunay edges, and removed edges longer than $\delta = 10$ Å in a postprocess. We refer to this representation of protein structure as DT graphs.

The definition of the Delaunay tessellation depends on the precise coordinate values given to its points, but we know that these coordinate values are not exact in the case of proteins due to measurement imprecision and atomic motions. Thus, Bandyopadhyay and Snoeyink recently defined the almost-Delaunay edges [1] by relaxing the empty sphere property to say that a pair of points p and q is joined by an almost-Delaunay edge with parameter ϵ, or $AD(\epsilon)$, if by perturbing all points by at most ϵ, p and q can be made to lie on an empty sphere. Equivalently, they look for a shell of width 2ϵ, formed by concentric spheres, so that p and q are on the outer sphere, and all points are outside the inner sphere. All Delaunay edges are in $AD(0)$, and $AD(\epsilon) \subseteq AD(\epsilon')$ for $\epsilon \leq \epsilon'$. Therefore, the almost-Delaunay edges are a superset of the Delaunay edges, whose size is controlled by the parameter ϵ. Various values of the parameter ϵ correspond to different allowed perturbations or motions. 0.1–0.25 Å would model decimal inaccuracies in the PDB coordinates or small vibrations, and 0.5–0.75 Å would model perturbations due to coarser motions. Thus, the protein graphs constructed with the almost-Delaunay edges are termed AD graphs. The precise parameter value for each edge of the contact distance graph can be computed by an algorithm that is much like the roundness algorithms from the computer-aided design (CAD) field of computational metrology. Code is available from http://www.cs.unc.edu/~debug/papers/AlmDel, or see [1] for algorithmic details.

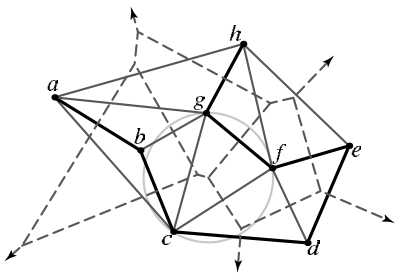

Figure 2: Examples of a Voronoi diagram and its dual Delaunay Tessellation for 2D points [a–f]

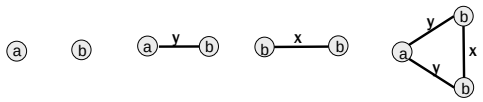

Figure 3: All frequent subgraphs (with support $\sigma = 2/3$) in $\mathcal{G}$ from Figure 1.

3. ALGORITHM DETAILS

In this section, we outline the framework we used to identify spatial motifs from a group of proteins. Those motifs can be used to classify protein families and identify protein family signatures, as further explained in the experimental study section. Our framework has two major components: (1) computing a graph representation for each protein as described above and (2) identifying significant common subgraphs from the database of protein graphs.

3.1 Mining Subgraphs From a Graph Database

3.1.1 Canonical Adjacency Matrix

We represent each graph by an adjacency matrix M such that every diagonal entry of M is filled with the label of the corresponding node and every off-diagonal entry is filled with the label of the corresponding edge, or zero if there is no edge. This is slightly different from the widely used adjacency matrix representation for unlabeled graphs, such as the one used by [5].

Given an $n \times n$ adjacency matrix M of a graph G with n nodes, we define the *code* of M, denoted by $code(M)$, as the sequence of lower triangular entries of M (including the entries at diagonal) in the order: $M_{1,1}M_{2,1}M_{2,2}...M_{n,1}M_{n,2}...M_{n,n-1}M_{n,n}$ where $M_{i,j}$ represents the entry at the ith row and jth column in M. Figure 4 shows examples of adjacency matrices and codes for the labeled graph P showing in Figure 1.

We use lexicographic order of sequences to define a total order over sequences. Given a graph G, its *canonical form* is the maximal code among all its possible codes. The adjacency matrix M which produces the canonical form is denoted as G's *canonical adjacency matrix* (CAM). For example, the adjacency matrix M_1 shown in Figure 4 is the CAM of the graph P from Figure 1, and $code(M_1)$ is the canonical form of the graph.

For a matrix N, we define the *proper maximal submatrix* (*submatrix* for short) as the matrix M obtained by removing the last row (and the symmetric entries in N) from N.

One valuable property of the canonical form we are using (compared to the forms of [17, 21]) is that, given a graph database $\mathcal{G}$, all frequent subgraphs (represented by their CAMs) can be organized into a rooted tree. This tree is referred to as the *CAM Tree* of $\mathcal{G}$ and is formally described as follows:

(i) The root of the tree is the empty matrix;

(ii) Each node in the tree is a distinct connected subgraph of G, represented by its CAM;

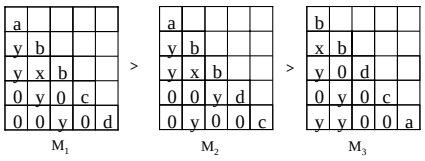

Figure 4: Three adjacency matrices for the graph P in Figure 1. After applying the total ordering, we have $code(M_1)$ = "$aybyxb0yxc00y0d$" $\geq$ $code(M_2)$ = "$aybyxb00yd0yx0c$" $\geq$ $code(M_3)$ = "$bxby0dxy0cyy00a$".

(iii) For a given none-root node (with CAM M), its parent is the graph represented by M's proper maximal submatrix.

3.1.2 Algorithm Overview

Below, we present a high-level outline of the frequent subgraph mining algorithm. Further details can be found in [14].

FFSM

1: $S \leftarrow$ { the CAMs of the frequent nodes }
2: $P \leftarrow$ { the CAMs of the frequent edges }
3: FFSM-Explore(P, S);

FFSM-Explore (P, S)

1: **for** $X \in P$ **do**
2: **if** $(X.isCAM)$ **then**
3: $S \leftarrow S \cup \{X\}$
4: $C \leftarrow$ { all matrices $M \mid X$ is submatrix of M}
5: remove CAM(s) from C that is either infrequent or not optimal
6: FFSM-Explore(C, S)
7: **end if**
8: **end for**

3.1.3 Post Processing using Mutual Information

Typically a large number of subgraphs are produced from the above mining procedure for a moderate support value. One may expect that many of these subgraphs may not be useful for subsequent tasks, such as classification. To select the most informative collection of subgraphs from the whole list, we used the information-theoretic metric of mutual information as follows.

We define a random variable X_G for a subgraph G in a graph database GD as

$$X_G = \begin{cases} 1 & \text{with probability } sup_G \\ 0 & \text{with probability 1-}sup_G \end{cases}$$

Given a graph G and its subgraph G', we define the mutual information $I(G, G')$ as
$I(G, G') = \sum_{X_G, X_{G'}} p(X_G, X_{G'}) \log_2 \frac{p(X_G, X_{G'})}{p(X_G)p(X_{G'})}$

where $p(X_G, X_{G'})$ is the (empirical) joint probability distribution of $(X_G, X_{G'})$, which is defined as

$$p(X_G, X_{G'}) = \begin{cases} sup_G & \text{if } X_G = 1,\ X_{G'} = 1 \\ 0 & \text{if } X_G = 1,\ X_{G'} = 0 \\ sup_{G'} - sup_G & \text{if } X_G = 0,\ X_{G'} = 1 \\ 1 - sup_{G'} & \text{otherwise} \end{cases}$$

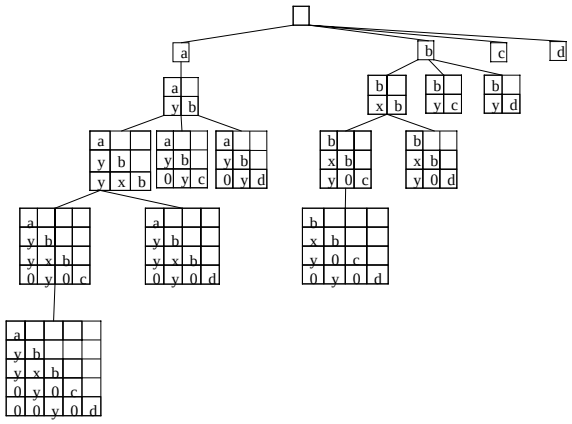

Figure 5: The CAM Tree for the frequent subgraphs in the graph P shown in Figure 1.

A pattern G is a *coherent subgraph* if the mutual information between G and each of its own subgraphs is above some threshold. Selecting only coherent subgraphs from the available frequent subgraph list offers several advantages: 1) it filters out subgraphs which are generic across families (for those subgraphs, the mutual information tends to be low) and (2) it finds statistically significant patterns since each coherent subgraph is strongly correlated with its own subgraphs. Our experimental study shows that coherent subgraph mining selects a small subset of features which have high distinguishing power between protein classes. Further details about coherent subgraph mining can be found in [15].

3.1.4 Classification Modeling

We built binary classification models using the Support Vector Machine (SVM) method [30]. There are several advantages of using SVM for the classification task in our context: 1) SVM is designed to handle sparse high-dimensional datasets (there are many features in the dataset and each feature may occur in only a small set of samples), and 2) there is a set of kernel learning functions (namely linear, polynomial and radius based) to choose from, depending on the property of the dataset. We used the `libsvm` program (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) for SVM classification and found the radius kernel worked best in our experiments.

4. RESULTS

4.1 Experimental Setup

All calculations were run on a single processor, 2.0GHz Pentium PC with 2GB memory, operating on RedHat Linux 7.3. The frequent subgraph mining algorithm was implemented using the C++ programming language and compiled using g++ with O3 optimization. We calculated the Delaunay tessellation for a set of coordinates using Quickhull [2]. The almost-Delaunay computation followed [1] using the code available at http://www.cs.unc.edu/~debug/papers/AlmDel/.

We obtained the coordinates for all proteins used in our studies from the Protein Data Bank (PDB). Three graph representations: CD, DT and AD were constructed for each protein using methods stated above. Figure 6 shows the average number of edges per vertex as a function of the threshold distance. For small distances the graphs are nearly the same, but as the distance grows, the number of edges in CD grows cubically, while DT remain almost constant and $AD(\epsilon)$ interpolates between CD and DT.

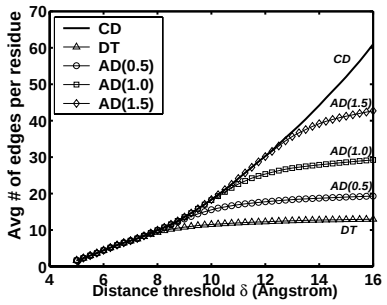

Figure 6: Average vertex degree as a function of the contact distance threshold for the three types of graphs representing Human Kallikrein 6 (1Lo6).

4.2 Binary Classification of SCOP families

Two datasets from the SCOP database [23] were used to evaluate the discrimination power of frequent subgraphs under a binary classification scheme. Proteins in the SCOP database are generally classified in five hierarchical levels: class, fold, superfamily, family and individual proteins. The first binary dataset (C_1) included two protein families that belong to two different SCOP classes. The first family is the nuclear receptor ligand-binding domain proteins (NB) from the all alpha class and the second one is the prokaryotic serine protease family (PSP) from the all beta class. The second dataset (C_2) included the families of eukaryotic serine proteases (ESP) and the prokaryotic serine proteases. These two families belong to the same superfamily. All the proteins included in the datasets C_1 and C_2 were selected from the *culled PDB* list (http://www.fccc.edu/research/labs/dunbrack/pisces/culledpdb.html) with no more than 60% pair-wise sequence similarity in order to remove highly homologous proteins. We retrieved proteins with resolution ≤ 3.0 and R factor ≤ 1.0 to ensure that we use high quality x-ray structures. The two datasets are further summarized in Table 1 below.

Data Set	Family I	size	Family II	size
C_1	NB	9	PSP	9
C_2	PSP	9	ESP	35

Table 1: Protein datasets

4.2.1 Protein Family Classification using Coherent Subgraph Counts as Variables

For each protein family we have identified common coherent subgraphs as discussed in Section 3.1.3. We used support thresholds ranging from 0.5 to 0.25; we report only the results with $\sigma = 0.3$, which gave the best classification accuracy. Each coherent subgraph with acceptable support was regarded as a feature, or variable, formally defined as follows.

Given n frequent subgraphs $f_1, f_2, \ldots, f_n$, we represent each protein G in a dataset as an n-element vector $V = v_1, v_2, \ldots, v_n$ in a feature space where v_i is the total number of distinct occurrences of the subgraph f_i in G.

For each feature f we have defined its *discrimination power* P as follows:

$$P = \left| \frac{f_{GA}}{S_A} - \frac{f_{GB}}{S_B} \right|,$$

where f_{GA} and f_{GB} are the total number of proteins in family A and B having f as a subgraph, and S_A and S_B are the size of family A and B, respectively. The greater the P value, the more selective the feature is.

A single classification experiment typically took less than ten minutes in our calculations. The classification results are summarized in Table 2. They demonstrate that the high classification accuracy was obtained using all three graph representations.

Data Set C_1	Features	Dist. Feat	Accuracy
DT	20,646	934	100%
AD(0.1)	23,130	1093	100%
AD(0.25)	26,943	1234	96%
AD(0.5)	32,463	1582	100%
AD(0.75)	37,394	1674	96%
CD	40,274	1859	95%
Data Set C_2	Features	Dist. Feat	Accuracy
DT	15,895	20	95%
AD(0.1)	18,491	29	95%
AD(0.25)	23,288	35	93%
AD(0.5)	29,083	35	95%
AD(0.75)	32,569	36	95%
CD	34,697	20	98%

Table 2: Binary classification results using DT, AD(ϵ), and CD protein graphs. Column two is the total number of features obtained. Column three is the number of distinguishing features selected to build the classification model. We use 0.75 as a threshold to select distinguishing features across all experiments. Column four lists the five-fold cross validation accuracy reported by the SVM program. Accuracy is defined for the test set as the fraction of true positives plus true negatives among all predicted.

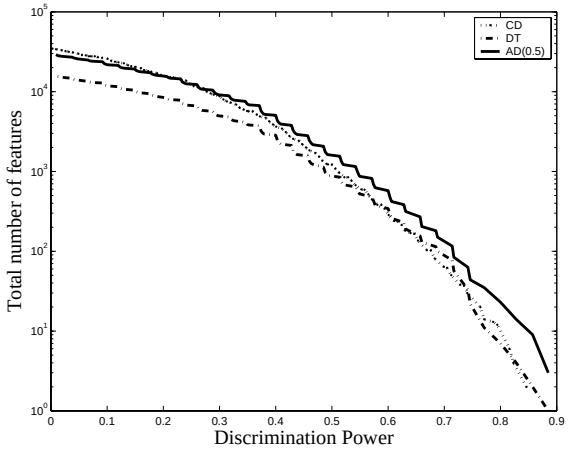

Figure 7: The total number of features extracted from the three graph representations of the dataset C_2 for different values of the discrimination power P.

For the dataset C_1, in which the two protein families are quite dissimilar, we expected to find a large number of features that distinguish the two families. Indeed we have obtained a large number of features with high discrimination power for all three graph representations (see Table 2).

For dataset C_2, in which the two families are quite similar to each other, we expected that only a handful of features will discriminate between the two families. The discrimination power of the features found for the three different graphs is shown in Figure 7. Interestingly, more features with high discrimination power were obtained with DT and AD graphs than with CD graphs, despite the fact that

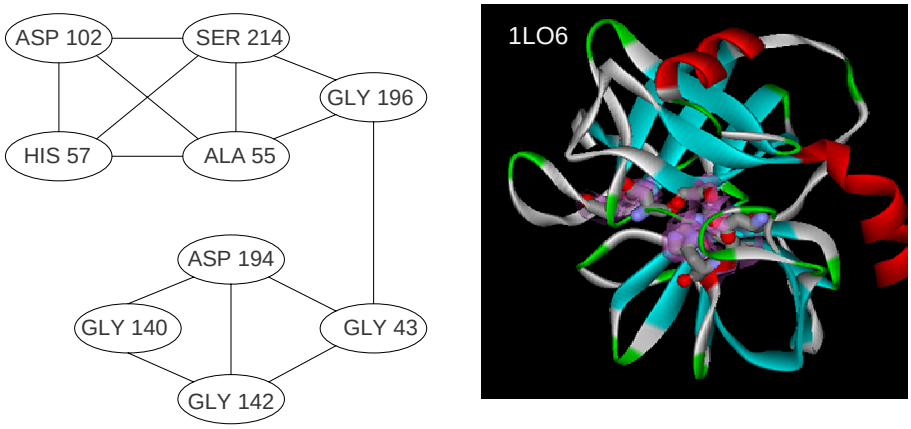

Figure 8: The largest subgraph motif found in every member of the Eukaryotic Protease Family. Left: graph representation. Right: mapping of this motif onto the backbone of Kallikrein 6 (1lo6). Note that this motif includes two members of the catalytic triad, i.e., His57 and Asp102.

the total number of features extracted from the latter graphs was the highest.

4.3 Signature Identification in Eukaryotic Serine Protease

Finding features (corresponding to packing motifs) that discriminate between the two protein families motivated us to further investigate the possibility of examining these motifs as characteristic *signatures* of a protein family. For a group P of proteins, represented by graphs, we define a signature of the group as a subgraph whose support in P is above certain high threshold (*minSupport*) whereas its support in the whole PDB database is less than some small upper bound (maxBackground). In our experimental study, we used minSuppport = 90% and maxBackground = 2%.

Using frequent subgraph mining with the DT graphs for dataset C_2, we obtained 3,298 features (subgraphs) which appeared in at least 90% of the members of the eukaryotic serine protease family. These calculations took about one minute to complete. There were 57 subgraphs with the background frequency as low as 0.6%. The background occurrence (or frequency) of a subgraph is defined as the number (or percent) of proteins found in a diverse subset of 500 proteins selected randomly from the 4800 proteins on the culled PDB list (excluding the Eukaryotic Serine Protease family) that contain the same subgraph. Thus, the background frequency of 0.6% implies that each feature was found in no more than 3 out of the 500 diverse random proteins. There were 438 subgraphs with background frequency less than 1%. Table 3 summarizes the background frequency distribution of the 2,086 subgraphs which have at most 2% background frequency.

λ	3	4	5	6
N	57	163	218	272
λ	7	8	9	10
N	302	330	361	383

Table 3: The background occurrence of features mined from the dataset C_2 using the DT graphs. λ: background occurrences, N, total number of features which have λ occurrences.

One of these features with the largest number of residues was chosen for more thorough examination. Figure 8 shows the corresponding subgraph and composition of this feature as well its mapping onto the structure of Kallikrein 6 (PDB code 1lo6). This feature includes two of the residues, His57 and Asp102, of the key ASP-HIS-SER triad from the catalytic site of serine proteases [31].

We notice an excessively long running time (>24 hours) if we use CD graphs representing the same protein dataset because of its dense representation. Frequent subgraph mining is known to be very sensitive to the density of graph database [14].

5. DISCUSSION AND FUTURE WORK

In this paper we report on the application of the frequent subgraph mining algorithm to protein structures represented as graphs. The goal of this investigation was to identify frequent subgraphs common to all (or the majority of) proteins belonging to the same structural and functional family in the SCOP database and explore these subgraphs as family specific amino acid residue signatures of the underlying family. Although protein graphs are complex, this application has become possible, thanks to several advanced features of the frequent subgraph mining algorithm [14, 15] employed in this paper.

Three graph representations of the protein structures termed CD, DT, and AD graphs have been explored to identify the most discriminatory subgraph-signatures. As discussed in Section 2.2, all three representations used the vertices defined by the proteins' C_α atoms, but they differed by the approach used to define the edges. These three representations have been compared in terms of the number and composition of the unique protein family specific features identified as the result of subgraph mining, computational efficiency of identifying these features, and the ability of these features to discriminate between SCOP families in the binary classification calculations using the SVM algorithm. Our results demonstrate that using highly simplified graph representations such as AD and DT graphs, we can still capture biologically meaningful signatures from SCOP families and as well as distinguishing features for classification purposes. The lists of family specific features identified from three different graph representations are not identical, suggesting that each graph representation captures unique aspects of protein structural organization. The total number of features was the highest as identified by the CD graphs followed by the AD graphs followed by the DT graphs. Nevertheless, the smaller number of features identified from AD graphs were shown to afford the highest discriminatory power in the binary classification experiments followed by the DT graphs. This result demonstrates that almost-Delaunay edges not only enrich the diversity of frequent subgraphs that could be identified by the FSM algorithm but are likely to capture additional functionally significant subgraphs that could not be detected by the Delaunay tessellation alone. We con-

clude that in order to achieve the highest accuracy in finding protein family specific signatures, AD graphs present the best choice both due to their relative computational efficiency and their robustness in taking into account possible experimental errors in determining protein atomic coordinates.

The success of this preliminary studies encourages us to consider several possible directions for future investigations of protein graphs with the FSM algorithm. We plan to develop an incremental subgraph mining algorithm that repeatedly increases the parameter ϵ in the AD until it has found the maximum number of the family specific signatures. We also plan to extend the classification experiments to multiple families using multi-class classification algorithms, rather than simple binary classification, with an ultimate goal of classifying the entire collection of SCOP families on the basis of family specific frequent subgraphs, or residue signatures. Finally, some of the individual subgraph-signatures identified as the result of protein structure analysis may also bear sequence specificity, i.e., contain characteristic residues found in the same order and approximately at the same separation distances in the underlying primary sequences of the protein family. Some instances of such structure-derived primary sequence motifs formally similar to PROSITE patterns [12] have been implicated in our earlier analyses of protein packing with Delaunay tessellation (recently summarized in [28]). This latter hypothesis offers an exciting new avenue in exploring structure-sequence-function relationships in proteins by using structure based subgraphs (residue patterns) for functional and structural annotation of not only novel proteins structures but also sequences resulting from the ongoing genomics projects.

6. ACKNOWLEDGEMENTS

This work was supported in part by NSF grants 9988742 and 0076984 awarded to JS, NSF grant MCB/ITR-0112896 and NIH grant GM066940-01 awarded to AT, and the bioinformatics graduate fellowship from UNC General Administration to LH.

7. REFERENCES

[1] D. Bandyopadhyay and J. Snoeyink. Almost-Delaunay simplices : Nearest neighbor relations for imprecise points. In *ACM-SIAM Symposium On Distributed Algorithms*, pages 403–412, 2004.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.

[3] C. Borgelt and M. R. Berhold. Mining molecular fragments: Finding relevant substructures of molecules. In *Proc. International Conference on Data Mining'02*.

[4] S. Chakraborty and S. Biswas. Approximation algorithms for 3-d common substructure identification in drug and protein molecules. *Workshop on Algorithms and Data Structures*, pages 253–264, 1999.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT press, 2001.

[6] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. In *4th International Conference on Knowledge Discovery and Data Mining*, pages 30–36, 1998.

[7] B. Delaunay. Sur la sphère vide. A la memoire de Georges Voronoi. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk*, 7:793–800, 1934.

[8] P. W. Finn, L. E. Kavraki, J. Latombe, R. Motwani, C. R. Shelton, S. Venkatasubramanian, and A. Yao. Rapid: Randomized pharmacophore identification for drug design. *Symposium on Computational Geometry*, pages 324–333, 1997.

[9] D. Fischer, H. Wolfson, S. L. Lin, and R. Nussinov. Three-dimensional, sequence order-independent structural comparison of a serine protease against the crystallographic database reveals active site similarities: potential implication to evolution and to protein folding. *Protein Science*, 3:769–778, 1994.

[10] H. Grindley, P. Artymiuk, D. Rice, and P. Willet. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *J. Mol. biol.*, 229:707–721, 1993.

[11] M. Hochsmann, T. Toller, R. G., and S. Kurtz. Local similarity in RNA secondary structures. In *Proc. Computational Systems Bioinformatics*, pages 159–168.

[12] S. K. Hofmann, P. Bucher, L. Falquet, and A. Bairoch. The prosite database, its status in 1999. *Nucleic Acids Res*, 27(1):215–219, 1999.

[13] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *Proc. International Conference on Data Mining'03*.

[14] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. *UNC computer science technical report TR03-021*, 2003.

[15] J. Huan, W. Wang, A. Washington, J. Prins, R. Shah, and A. Tropsha. Accurate classification of protein structural families based on coherent subgraph analysis. In *Proc. Pacific Symposium on Biocomputing*, 2004.

[16] P. Indyk, R. Motwani, and S. Venkatasubramanian. Geometric matching under noise: Combinatorial bounds and algorithms. *ACM Symposium on Discrete Algorithms*, 1999.

[17] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *In PKDD'00*.

[18] D. Jacobs, A. Rader, L. Kuhn, and M. Thorpe. Graph theory predictions of protein flexibility. *Proteins: Struct. Funct. Genet.*, 44:150–155, 2001.

[19] P. N. Klein. Computing the edit-distance between unrooted ordered trees. In *Proc. 6th Annual Enropean Symposium*, pages 91–102, 1998.

[20] G. J. Kleywegt. Recognition of spatial motifs in protein structures. *Journal of Molecular Biology*, 285(4):1887–1897, 1999.

[21] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proc. International Conference on Data Mining'01*.

[22] J. Liang, H. Edelsbrunner, P. Fu, P. Sudhakar, and S. Subramaniam. Analytical shape computing of macromolecules I: molecular area and volume through alpha shape. *Proteins*, 33:1–17, 1998.

[23] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–40, 1995.

[24] F. M. Richards. The interpretation of protein structures: total volume, group volume distributions, and packing density. *J. Molecular Biology*, 82:1–14, 1974.

[25] B. Shapiro. An algorithm for comparing multiple RNA secondary structures. *Comp. Appl. Biosci*, 4(3):387–393, 1988.

[26] R. Singh, A. Tropsha, and I. Vaisman. Delaunay tessellation of proteins. *J. Comput. Biol.*, 3:213–222, 1996.

[27] H. T. T. Akutsu and T. Tokuyama. Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. In *13th Annual ACM Symp. on Computational Geometry*, pages 314–323, 1997.

[28] A. Tropsha, C. Carter, S. Cammer, and I. Vaisman. Simplicial neighborhood analysis of protein packing (SNAPP) : a computational geometry approach to studying proteins. *Methods Enzymol.*, 374:509–544, 2003.

[29] J. Tsai, R. Taylor, C. Chothia, and M. Gerstein. The packing density in proteins: Standard radii and volumes. *Journal of Molecular Biology*, 290(1):253–266, 1999.

[30] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.

[31] S. Vishveshwara, K. V. Brinda, and N. Kannan. Protein structure: Insights from graph theory. *J. of Theo. and Comp. Chem.*, 1(1):187–211, 2002.

[32] H. Wako and T. Yamato. Novel method to detect a motif of local structures in different protein conformations. *Protein Engineering*, 11:981–990, 1998.

[33] J. Wang, B. Shapiro, D. Sasha, K. Zhang, and K. M. Currey. A algorithm for finding the largest approximately common substructures of two trees. *IEEE Transactions on Pattern Anaylsis and Machine Intelligence*.

[34] L. Wernisch, M. Hunting, and S. Wodak. Identification of structural domains in proteins by a graph heuristic. *Proteins*, 35(3):338–352, 1999.

[35] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proc. International Conference on Data Mining'02*.

[36] K. Zhang, R. Stgatman, and D. Shasha. Simple fast algorithm for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.