

Mining Spatial Motifs from Protein Structure Graphs

Jun Huan¹, Wei Wang¹, Deepak Bandyopadhyay¹, Jack Snoeyink¹, Jan Prins¹, Alex Tropsha²

¹ Department of Computer Science, University of North Carolina, 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, Chapel Hill
tropsha@email.unc.edu

Abstract

Finding recurring structural features among proteins three-dimensional (3D) structures is an important problem in bioinformatics. In this paper we apply a novel subgraph mining algorithm to three related graph representations of the sequence and proximity characteristics of a protein's amino acid residues. The subgraph mining algorithm is used to discover spatial motifs that can be used to discriminate among proteins in different families found in the SCOP database.

The results indicate that an Delaunay Tessellation (and its recent developed extension almost-Delaunay) subset of the contact graph is robust, sparse, and adequate to produce asymptotically simplified graphs (with increasing interaction radius) for mining spatial motifs, yielding motifs with discrimination qualities similar to, or better than, those obtained from the full contact graph.

1 Introduction

1.1 Spatial Motif Discovery in Proteins

Recurring structures in proteins reveal important information about protein structure and function. Common structural fragments of various sizes have fixed 3D arrangements of residues that correspond to active sites or other functionally relevant features, such as Prosite patterns [12]. Identifying such *spatial motifs* in proteins may have great impact on protein classification [5], protein function prediction [10] and protein folding [19].

Protein structure may be modeled using a variety of graph representations [28]. Our approach uses a labeled graph in which the nodes represent the amino-acid residues comprising the protein, and the edges represent proximity relations among the residues. Two types of edges are iden-

tified: a bond edge connects two residues that are contiguous in the primary sequence, and a proximity edge connects two (non-bonded) residues within a given distance δ of each other. Spatial motifs appear as recurring subgraphs among a set of proteins represented in this fashion.

Several algorithms have been developed recently in the data mining community to find all frequent subgraphs of a group of general graphs [20, 31, 13, 14]. The techniques have been successfully applied to cheminformatics by modeling chemical compounds with undirected graphs. Recurring substructures in a group of chemical compounds with similar activity are identified by finding frequent subgraphs in their related graphical representations. The recurring substructures can indicate chemical features characterizing the compounds activities [7, 4].

Applying frequent subgraph mining to find patterns from a group of proteins is non-trivial. As we reported in our prior work [13], the total number of frequent subgraphs for a set of graphs grows exponentially as the graph size increases. 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). As 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.

In this paper we investigate techniques to reduce the number of edges in a contact graph (CG). In particular, we choose edges from the Delaunay tessellation and its recently developed extension to almost-Delaunay [1]. The Delaunay tessellation (DT), defined below, captures neighbor relations between points representing residues or atoms. It has been used to analyze packing [23, 26] and structure [21, 24, 30, 29] in proteins. The almost-Delaunay edges (AD) expand the set of Delaunay edges to account for perturbation or motion of point coordinates, controlled by a parameter ϵ . A property of these representations is that $DT \subseteq AD(\epsilon) \subseteq CG$ for all $\epsilon \geq 0$.

Our experimental study demonstrates that using AD we

significantly reduced the graph size without degrading the quality of the mined features as measured by their accuracy when used in SCOP protein-family binary classification experiments, and by their ability to find distinguishing features that are highly specific to single families of proteins.

1.2 Related Work

Several research groups have addressed the problem of finding spatial motifs by using computational geometry/computer vision approaches. A protein can be represented as a set of points in the R^3 space and the problem of (pair-wise) spatial motif finding may be formalized as the Largest Common Pointset (LCP) problem: identifying the largest common subset of two sets of points [25]. A number of variations to this problem have been explored, which include approximate LCP problem [5, 16] and LCP- α : identifying a common point set S that has size with an approximation factor α to the maximal set [9].

Because of the expressive power of graphs to represent complicated data, graph theory has been used to study protein structures, including active site clusters, folding clusters, aromatic clusters/thermal stability, and protein-protein interaction. See [28] for a recent and comprehensive review on applying graph theory to protein structure analysis. The choice of graph representation is a key issues in applying graph-related techniques to analyze protein structures. Several representations have been developed, ranging from coarse representations in which each secondary segment is a node [11] to fine representations in which each atom is a node [18].

The remainder of the paper is organized as follows. Section 2 presents definitions for subgraph isomorphism, Delaunay tessellation and almost-Delaunay edges. Section 3 presents the data structure and the algorithm for subgraph mining and Section 4 present our experimental study of protein classification.

2 Background

2.1 Labeled Graph

We defined a *labeled graph* G as a four element tuple $G = \{V, E, \Sigma, l\}$ where V is the set of nodes of G and $E \subseteq V \times V$ is a set of undirected edges of G . We have a label set Σ that we assume has a total order \geq . Labels are assigned by a labeling function $l: V \cup E \rightarrow \Sigma$. The same label may appear on multiple nodes or on multiple edges, but we require that the set of edge and set of node labels are disjoint.

A labeled graph $G = (V, E, \Sigma, l)$ is *isomorphic* to another graph $G' = (V', E', \Sigma', l')$ iff there is a bijection that

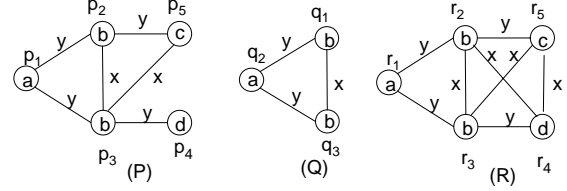


Figure 1. Example of a graph database GD of three labeled graphs with an induced subgraph isomorphism. We assume the node and edge labels are ordered s.t. $a > b > c > x > y$. The mapping $q_1 \rightarrow p_2, q_2 \rightarrow p_1, q_3 \rightarrow p_3$ represents an induced subgraph isomorphism from graph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

preserves labels $f: V \rightarrow V'$ such that:

$$\begin{aligned} \forall u \in V, l(u) &= l'(f(u)) \\ \forall u, v \in V, \left((u, v) \in E \iff (f(u), f(v)) \in E' \right) \\ \wedge l(u, v) &= l'(f(u), f(v)) \end{aligned}$$

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

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

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 GD (referred to as a *graph database*), the *support* of a graph G is defined as the fraction of graphs in GD of which G is a subgraph. For each graph database GD , we choose a threshold $0 < t \leq 1$, and say that G is *frequent* iff its *support* is at least threshold t (denoted as *minSupport*). The problem of *Frequent Subgraph Mining* is to identify all frequent subgraphs of GD . Figure 3 represents all the frequent subgraphs w.r.t. $\text{minSupport} = 2/3$ for the set of graphs in Figure 1.

2.2 Delaunay and almost-Delaunay

We use the following definitions in choosing the edges of the graphical representation of a protein. The Delaunay tessellation [8] is defined for a finite set of points by an *empty*

sphere property: A pair of points is joined by an edge if and only if 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 if and only if their Voronoi cells share a common face. Figure 2 illustrates the Delaunay in 2D with solid lines, and the dual Voronoi with dashed.)

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. Thus, Bandyopadhyay and Snoeyink recently defined the almost-Delaunay edges [1, 2] by relaxing the empty sphere property to say that an edge pq is almost-Delaunay with parameter ϵ , or $AD(\epsilon)$, if by perturbing all points by at most ϵ , edge pq can be made to lie on an empty sphere. Equivalently, they look for a shell of width 2ϵ , formed by concentric spheres, so that edge pq is inside the outer sphere, and all points are outside the inner sphere. Since all Delaunay edges are in $AD(0)$, and $AD(\epsilon) \subseteq AD(\epsilon')$ for $\epsilon \leq \epsilon'$, the almost-Delaunay edges are a superset of the Delaunay edges whose size is controlled by the threshold parameter ϵ . Various values of the threshold parameter 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.

3 Algorithm Details

In this section, we outline the framework we used to identify spatial motifs from a group of proteins. Those motifs will be used to classify proteins and to identify protein family signatures, as further explained in the experimental study section.

Our framework has two major steps: (1) computing a graphical representation for each protein using Delaunay tessellation or almost-Delaunay edges and (2) identifying

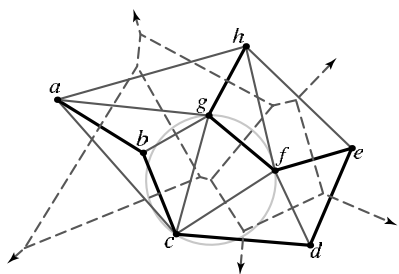


Figure 2. Examples of a Voronoi diagram and its dual Delaunay Tessellation for 2D points [a–f]

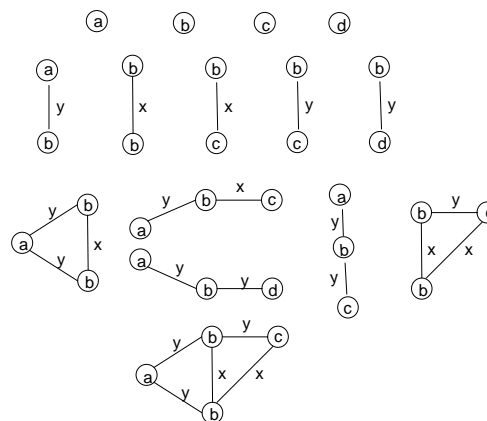


Figure 3. all frequent subgraphs (with $minSupport = 2/3$) in GD in Figure 1.

significant patterns from the database of graphical representations.

3.1 Building Protein Graphs

Because we are interested in structures exhibited by the protein backbone, we take the C_α atom coordinates as the nodes of our protein graphs.

Because we are interested in residues that interact, we restrict our edges to those that have length at most 10 Å. The set of all such edges, which we call the *contact graph*, can be determined by bucketting. At the other extreme, we can select only those edges of the Delaunay tessellation that are less than 10 Å, because these edges represent neighbor relationships that are not shielded by other residues. We use quickhull [3] to compute the tessellation edges. The almost-Delaunay edges interpolate between these two extremes, giving the Delaunay edges for parameter 0 and the complete set of contact edges for some parameter bounded above by 5 Å. The precise parameter value for each edge of the contact 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.

3.2 Mining Subgraphs From a Graph Database

3.2.1 Canonical Adjacency Matrix

In the following discussion, we lay out the formal base for mining subgraphs from a graph database. 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

where $\text{sup}(G)$ is the support of the subgraph G in the database.

Given a graph G and its subgraph G' , we define the mutual information $I(G, G')$ as follows:

$$I(G, G') = - \sum_{x \in X_G, y \in X_{G'}} p(x, y) \log_2 \frac{p(x, y)}{p(x) \times p(y)}.$$

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

$$X_G, X_{G'} = \begin{cases} (1, 1) & \text{with probability } \text{sup}(G) \\ (1, 0) & \text{with probability } 0 \\ (0, 1) & \text{with probability } \text{sup}(G') - \text{sup}(G) \\ (0, 0) & \text{with probability } 1 - \text{sup}(G') \end{cases}$$

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 will tend 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 coherent subgraph mining selects a small subset of features which have high distinguishing power between classes. Further details about coherent subgraph mining can be found in [15].

4 Experimental Study

We performed our experimental study using a single processor of a 2.0GHz Pentium PC with 2GB memory, running RedHat Linux 7.3. The frequent subgraph mining algorithm is implemented using the C++ programming language and compiled using g++ with O3 optimization. We calculate the Delaunay Tessellation for a set of coordinates using the related function provided by Matlab. The almost-Delaunay computation follows [1] using the code available at <http://www.cs.unc.edu/debug/papers/AlmDel/>.

4.1 Building Graphical Representations of Proteins

For each protein in our experimental study, we extract C_α coordinates from the PDB database (<http://www.rcsb.org/pdb/>) and build three types of labeled undirected graphs.

The *contact graph* (CG) has a labeled node for each amino acid residue. Two residues i, j have a *bond edge* if they are joined by a peptide bond, or a *proximity edge* if the distance between the two associated C_α atoms is less than some threshold δ . In the *Delaunay Tessellation* (DT) we retain only those proximity edges of the CG that satisfy the Delaunay empty sphere criterion. These are the edges of the

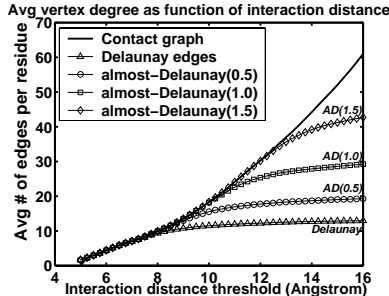


Figure 6. Vertex degrees for the three types of graphs constructed from Eukaryotic Serine Protease for increasing interaction distance thresholds.

Delaunay graph whose length is less than δ . For the *almost-Delaunay* (AD) with parameter ϵ , we retain proximity edges of the CG that satisfy the almost-Delaunay criterion.

For a given interaction distance threshold δ , we have that $DT \subseteq AD(\epsilon) \subseteq CG$ for any $\epsilon \geq 0$. In Figure 6 we show the average degree of these graph representations for the Eukaryotic Serine Protease as a function of the interaction distance threshold. From the figure we can see that if we consider only short range interactions ($\leq 8 \text{ \AA}$), the three graph representations are almost the same. However, as we increase the interaction distance threshold, the total number of edges in CG increases rapidly. Volume packing considerations suggest that the average degree can rise with the cube of the interaction radius, which was borne out by our data set (details not shown here). In contrast, the DT graph hardly varies with interaction distance. The AD graph smoothly shifts between the DT and CG extremes for increasing perturbation value ϵ .

For subgraph mining, our ideal graph would have labeled edges between almost all interacting residues, and few between non-interacting residues. We expected that the contact graph (CG) would satisfy the former, but not the latter, since it will contain proximity edges between residues whose interaction is obscured by other residues in between. We hoped that the Delaunay (DT) would capture only the essential interactions, but feared that its precise geometric definition, even if applied to two biologically identical protein structures, would discriminate mathematical differences in the protein coordinates assigned due to differences in crystallographic resolution, refinement, and quality parameters. We therefore included the almost-Delaunay edges for parameter values from 0.1 to 0.75 \AA , to capture edges more robustly under perturbation and motion of residue coordinates.

4.2 Binary Classification using SCOP families

Two datasets from the SCOP database [22] were used to evaluate the performance of the proposed algorithm under a binary (pairwise) classification scheme. SCOP is a database maintained by a domain expert to hierarchically classify proteins in five levels: Class, Fold, Superfamily, Family and individual proteins. Our first dataset (C_1) includes two protein families that belong to two different SCOP classes. The first family is the Nuclear receptor ligand-binding domain family from the all alpha proteins class and the second one is the Prokaryotic Serine Protease family from the all beta proteins class. Our second dataset (C_2) included all proteins of the Eukaryotic Serine Protease family (family 1) and proteins of the Prokaryotic Proteases family (family 2). These two protein families belong to the same superfamily. All the proteins in the dataset are selected from the *culled PDB* list (<http://www.fccc.edu/research/labs/dunbrack/pisces/culledpdb.html>) with less than 60% sequence homology in order to remove redundancies from the datasets. We retrieve proteins with resolution ≤ 3.0 and R factor ≤ 1.0 to ensure that we use high quality xray structures. The two datasets are further summarized in Table 1 below.

Data Set	C_1	C_2
Family 1 size	13	35
Family 2 size	9	9

Table 1. Dataset Summary

4.2.1 Feature Extraction Using Coherent Subgraph Mining

For each of the datasets, we combined all proteins from the two families and ran the frequent subgraph mining algorithm followed by the coherent subgraph post-process technique [15] to retrieve all the coherent subgraphs as features. In our experimental study, we use support thresholds ranging from 0.5 to 0.25; we report only the results with threshold 0.3, which gave the best classification accuracy.

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

For dataset C_1 , in which the two protein families are quite dissimilar to each other, we would expect to find a large number of features separating the two families, and this is borne out in the classification results of the next section (see Table 2). Any of the graph representations work to obtain features with high discrimination power for this data set.

For dataset C_2 , in which the two families are quite similar to each other, we expect only a handful of features will separate the two families. The discrimination power of the features found for the three different graphs is shown in Figure 8. The DT and AD graphs obtain more features with high discrimination power despite the fact that many fewer features overall were mined from these graphs.

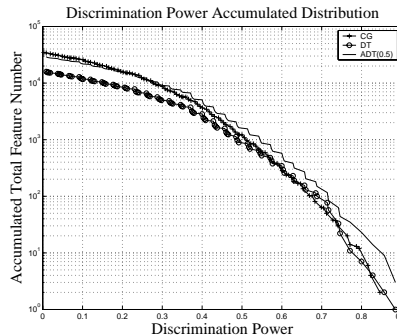


Figure 7. The accumulated discrimination power of mined features for dataset C_2 .

4.2.2 Classification Result

We built binary classification models using the Support Vector Machine (SVM) method [27]. 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). 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, which is downloadable from <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Mining features from the datasets in our current experiments usually takes less than ten minutes without little variation. The experimental results show that we are able to obtain high classification accuracy using all three graphical representations. We noticed that AD (including DT) always mines less features when comparing to CG but still captures those with high discrimination powers. The classification results are further summarized in Table 2.

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
CG	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
CG	34,697	20	98

Table 2. Binary Classification Results different graph representations: DT (Delaunay) and AD (almost-Delaunay, with perturbation in the parenthesis), CG (contact graph). Column two is the total number of features obtained. Column three is the number of distinguishing features selected to build the classification. We use the threshold 0.75 to select distinguishing features across all experiments. Column four lists the five-fold cross validation accuracy reported by the SVM program. Accuracy is defined as the fraction of true positive and true negative in the test set.

4.3 Signature Identification in Eukaryotic Serine Protease

Finding features that distinguish the two proteins families motivated us to further investigate the possibility of identifying *signatures* of a protein family. For a group P of proteins, represented by their graphical representations, we define a signature of the group as a subgraph whose support in P is at least some threshold ($minSupport$) and its support in the whole PDB database is less than some upper bound ($maxBackground$). In our experimental study, we used $minSupport = 90\%$ and $maxBackground = 2\%$.

Using frequent subgraph mining with the DT graph for dataset C_2 , we obtained 3,298 features which appear in at least 90% of the members in the Eukaryotic Serine Protease family. Out of these features, there are 57 subgraphs that have background frequency as low as 0.6% (3 out of the 500 proteins) and a total of 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% frequency in the background.

The subgraph mining algorithm took around one minute to find the 3,298. The same search on the contact graph was aborted when the CPU time exceeded 12 hours (for AD graphs, it took up to six minutes to complete the task).

From these features, we selected the one with the largest number of residues. Using existing knowledge about serine

protease family [28], we found that this feature contains the ASP-HIS-SER triad which matches the active center of serine protease. Figure 8 shows the corresponding subgraph position within the backbone of Kallikrein 6 (1lo6).

5 Discussion and Future Work

We find that we may substitute mining the DT or AD graphs for the CG without significant loss of features found for discrimination tasks or in identifying signatures for Protein families. At the same time the computational complexity is substantially reduced when using the DT or AD.

The neighbor relationships between residues are well captured by the DT, so that graphs with DT edges can already be used to identify significant protein structure. When comparing across many proteins, however, the AD should be used to ensure that the resulting graph is robust in the face of protein motion and experimental error in determining coordinate positions.

Some directions for future work include extending the classification experiments to multiple families, rather than simple binary classification. Classification across the entire SCOP database is an interesting goal, especially when the motifs upon which such a classification rests can be elucidated.

Another direction of work is to develop an incremental subgraph mining algorithm that repeatedly increases the parameter ϵ in the AD until it has found sufficient features for a classification or signature identification task.

References

- [1] D. Bandyopadhyay and J. Snoeyink. Almost-Delaunay simplices : Robust neighbor relations for imprecise points. In *Symposium On Distributed Algorithms*, 2004. to appear.
- [2] D. Bandyopadhyay, A. Tropsha, and J. Snoeyink. Analyzing protein structure with almost-delaunay tetrahedra, 2004. Submitted to RECOMB'04. <http://www.cs.unc.edu/~debug/papers/AlmDel>.
- [3] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.
- [4] C. Borgelt and M. R. Berhold. Mining molecular fragments: Finding relevant substructures of molecules. In *ICDM'02*.
- [5] 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 2534–264, 1999.
- [6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT press, 2001.
- [7] L. Dehaspe, H. Toivonen, and R. D. King. Finding frequent substructures in chemical compounds. *Proc. of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 30–6, 1998.

Background occurrence	3	4	5 (1%)	6	7	8	9	10 (2%)
Number of features	57	163	218	272	302	330	361	383

Table 3. Number of features mined from dataset C_2 using the DT graph that have a given background occurrence among a set of 500 proteins sampled from the 4800 on the culled PDB list, excluding the Eukaryotic Serine Protease family.

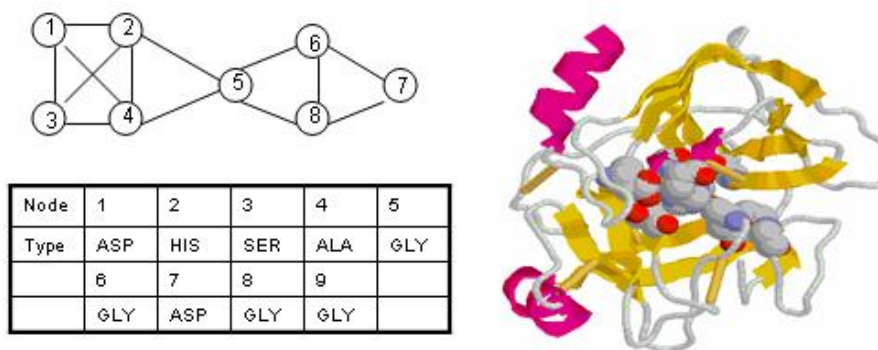


Figure 8. A motif that exists in every member of the Eukaryotic Protease Family. Left: graph representation. Right: occurrences within the backbone of Kallikrein 6 (11o6).

- [8] B. Delaunay. Sur la sphère vide. A la memoire de Georges Voronoi. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.
- [9] P. W. Finn, L. E. Kavasaki, J. Latombe, R. Motwani, C. R. Shelton, S. Venkatasubramanian, and A. Yao. Rapid: Randomized pharmacophore identification for drug design. *Symposium on Computational Geometry*, pages 324–33, 1997.
- [10] 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 Sci.*, 3:769–78, 1994.
- [11] 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.*, 229:707–21, 1993.
- [12] S. K. Hofmann, P. Bucher, L. Falquet, and A. Bairoch. The prosite database, its status in 1999. *Nucleic Acids Res*, 27(1):215–9, 1999.
- [13] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. *UNC computer science technique report TR03-021*, 2003.
- [14] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. in *ICDM'03*, 2003. to appear.
- [15] J. Huan, W. Wang, A. Washington, J. Prins, and A. Tropsha. Accurately classify protein family based on coherent subgraph mining. in *Pacific Symposium on Biocomputing*, 2004. to appear.
- [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] G. J. Kleywegt. Recognition of spatial motifs in protein structures. *J Mol Biol.*, 285(4):1887–97, 1999.
- [20] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *ICDM'01*.
- [21] J. Liang, H. Edelsbrunner, P. Fu, P. Sudhakar, and S. Subramanian. Analytical shape computing of macromolecules I: molecular area and volume through alpha shape. *Proteins*, 33:1–17, 1998.
- [22] A. Murzin, S. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247:536–40, 1995.
- [23] F. M. Richards. The interpretation of protein structures: total volume, group volume distributions, and packing density. *J. Molecular Biology*, 82:1–14, 1974.
- [24] R. Singh, A. Tropsha, and I. Vaisman. Delaunay tessellation of proteins. *J. Comput. Biol.*, 3:213–222, 1996.
- [25] 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 *Proc. 13th Annual ACM Symp. on Computational Geometry*, pages 314–23, 1997.
- [26] 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.
- [27] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.
- [28] 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.

- [29] H. Wako and T. Yamato. Novel method to detect a motif of local structures in different protein conformations. *Protein Engineering*, 11:981–990, 1998.
- [30] L. Wernisch, M. Hunting, and S. Wodak. Identification of structural domains in proteins by a graph heuristic. *Proteins*, 35(3):338–352, 1999.
- [31] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. *In ICDM'02*.