Accurate Classification of Protein Structural Families Using Coherent Subgraph Analysis

J. Huan[1], W. Wang[1], A. Washington[1], J. Prins[1], R. Shah[2], A. Tropsha[2†]
[1]*Department of Computer Science,*
[2]*The Laboratory for Molecular Modeling, Division of Medicinal Chemistry and Natural Products, School of Pharmacy,*
University of North Carolina, Chapel Hill, NC 27599

Protein structural annotation and classification is an important problem in bioinformatics. We report on the development of an efficient subgraph mining technique and its application to finding characteristic substructural patterns within protein structural families. In our method, protein structures are represented by graphs where the nodes are residues and the edges connect residues found within certain distance from each other. Application of subgraph mining to proteins is challenging for a number reasons: (1) protein graphs are large and complex, (2) current protein databases are large and continue to grow rapidly, and (3) only a small fraction of the frequent subgraphs among the huge pool of all possible subgraphs could be significant in the context of protein classification.

To address these challenges, we have developed an information theoretic model called *coherent subgraph* mining. From information theory, the entropy of a random variable X measures the information content carried by X and the Mutual Information (MI) between two random variables X and Y measures the correlation between X and Y. We define a subgraph X as coherent if it is strongly correlated with every sufficiently large sub-subgraph Y embedded in it. Based on the MI metric, we have designed a search scheme that only reports coherent subgraphs.

To determine the significance of coherent protein subgraphs, we have conducted an experimental study in which all coherent subgraphs were identified in several protein structural families annotated in the SCOP database (Murzin et al, 1995). The Support Vector Machine algorithm was used to classify proteins from different families under the binary classification scheme. We find that this approach identifies spatial motifs unique to individual SCOP families and affords excellent discrimination between families.

1 Introduction

1.1 *Spatial Motif Discovery in Proteins*

Recurring substructures in proteins reveal important information about protein structure and function. For instance, common structural fragments may represent

† This work supported in part by the NSF grant MCB/ITR-0112896 to AT, a bioinformatics graduate fellowship from UNC General Administration to LH, and a UNC Summer Pre-Graduate Research Experience award to AW.

fixed 3D arrangements of residues that correspond to active sites or other functionally relevant features such as Prosite patterns (Hofmann, et al. 1999). Understanding recurring substructures in proteins aids in protein classification (Chakraborty et al. 1999), function prediction (Fischer et al. 1994), and folding (Kleywegt 1999).

Many computational methods have been proposed to find motifs in proteins. Multiple sequence alignments of proteins with similar structural domains (Henikoff, et al 1999) could be used to provide information about the possible common substructures in the hope that conserved sequence patterns in a group of homologous proteins may have similar 3D arrangements. This method generally doesn't work very well for proteins that have low sequence similarity although structurally similar proteins can have sequence identities below 10%, far too low to propose any structural similarity on the basis of sequence comparison (Orengo & Taylor, 1996).

Several research groups have addressed the problem of finding spatial motifs by using computational geometry/computer vision approaches. From the geometric point of view, a protein can be modeled as a set of points in the R^3 space and the problem of (pairwise) spatial motif finding can be formalized as that of finding the Largest Common Point (LCP) set. (Akutsu et al. 1997). Plenty of variations to this problem have been explored, which include approximate LCP problem (Chakraborty et al. 1999, Indyk et al. 1999) and LCP-α (finding a sufficiently large common point set S of two sets of points but not necessarily the maximal one) (Finn et al. 1997).

Applying frequent subgraph mining techniques to find patterns from a group of proteins is a non-trivial task. The total number of frequent subgraphs for a set of graphs grows exponentially as the average graph size increases, as graphs become denser, as the number of node and edge labels decreases and as the size of the recurring subgraphs increases (Huan et al 2003). For instance, for a moderate protein dataset (about 100 proteins with the average of 200 residues per protein), the total number of frequent subgraphs could be extremely high (>> one million). Since the underlying operation of subgraph isomorphism testing is NP-complete, it is critical to minimize the number of frequent subgraphs that should be analyzed.

In order to apply the graph based spatial motif identification method to proteins, we have developed a novel information theoretic model called *coherent subgraphs.* A graph G is coherent if it is strongly correlated with every sufficiently large subgraph embedded in it. As discussed in the following parts of this report, coherent subgraphs capture discriminative features and afford high accuracy of protein structural classification.

1.2 Related Work

Finding patterns from graphs has long been an interesting topic in the data mining/machine learning community. For instance, Inductive Logic Programming (ILP) has been widely used to find patterns from graph dataset (Dehaspe 1998).

However, ILP is not designed for large databases. Other methods focused on approximation techniques such as SUBDUE (Holder 1994) or heuristics such as greed based algorithm (Yoshida and Motoda, 1995). Several algorithms have been developed in the data mining community to find all frequent subgraphs of a group of general graphs (Kuramochi and Karypis 2001, Yan and Han 2002, Huan et al. 2003). These techniques have been successfully applied in cheminformatics where compounds are modeled by undirected graphs. Recurring substructures in a group of chemicals with similar activity are identified by finding frequent subgraphs in their related graphical representations. The recurring substructures can implicate chemical features responsible for compounds' biological activities (Deshpande et al. 2002).

Recent subgraph mining algorithms can be roughly classified into two categories. Algorithms in the first category use a level-wise search scheme like Apriori (Agrawal and Srikant, 1994) to enumerate the recurring subgraphs. Examples of such algorithms include AGM (Inokuchi et al. 2000) and FSG (Kuramochi and Karypis 2001). Instead of performing the level-wise search scheme, algorithms in the second category use a depth-first enumeration for frequent subgraphs (Yan and Han 2002, Huan et al. 2003). A depth-first search usually has better memory utilization and thus better performance. As reported by Yan and Han (2002), a depth-first search, can outperform FSG, the current state-of-the-art level-wise search scheme by an order of magnitude overall.

All of the above methods rely on a single threshold to qualify interesting patterns. Herein, we propose the coherent subgraph model using a statistical metric to qualify interesting patterns. This leads to more computationally efficient yet more accurate classification.

The remaining part of the paper is organized as follows. Section 2 presents a formal base for the coherent subgraph mining problem. This includes the definition of the labeled graph and labeled graph database (Section 2.1), the canonical representation of graphs (Section 2.2), the coherent subgraph mining problem, and our algorithm for efficient coherent subgraph mining (Section 2.3). Section 3 presents the results of an experimental study to classify protein structural families using the coherent subgraph mining approach and a case study of identifying fingerprints in the family of serine proteases. Finally, Section 4 summarizes our conclusions and discusses future challenges.

2 Methodology

2.1 Labeled Graph

We define a *labeled graph* G as a four element tuple $G = \{V, E, \sum, l\}$ where V is the set of nodes of G and $E \subseteq V \times V$ is the set of undirected edges of G. $\sum$ is a set of labels and the labeling function $l: V \cup E \rightarrow \sum$ maps nodes and edges in G to their

labels. The same label may appear on multiple nodes or on multiple edges, but we require that the set of edge labels and the set of node labels are disjoint. For our purposes we assume that there is a total order ≥ associated with the label set Σ.

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

$\forall$ u $\in$ V, l(u) = l'(f(u)), and

$\forall$ u, v $\in$ V, (((u,v) $\in$ E $\Leftrightarrow$ (f(u), f(v)) $\in$ E') $\wedge$ l(u,v) = l'(f(u), f(v))).

The bijection f denotes an *isomorphism* between G and G'.

A labeled graph G= (V, E, Σ, l) is an *induced subgraph* of graph G'=(V',E', Σ', l') iff

$V \subseteq V'$,

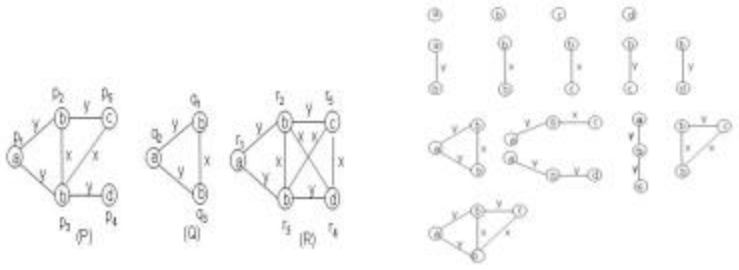
$E \subseteq E'$,

$\forall$ 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". Examples of labeled graphs, induced subgraph isomorphism, and frequent induced subgraphs are presented in Figure 1.



a b

Figure 1. (a): Examples of three labeled graphs (referred to as a graph database) and an induced subgraph isomorphism. The labels of the nodes are specified within the circle and the labels of the edges are specified along the edge. We assume the order a > b > c > d > x > y > 0 throughout this paper. 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. (b) All the frequent induced

Given a set of graphs GD (referred to as a *graph database*), the *support* of a graph G, denoted by sup_G is defined as the fraction of graphs in GD which embeds the subgraph G. Given a threshold t ($0 < t \leq 1$) (denoted as *minSupport*), we define G to be frequent, iff sup_G is at least t. All the frequent induced subgraphs in the graph database GD presented in Figure 1 (a) (with minSupport 2/3) are presented in Figure 1 (b).

Throughout this paper, we use the term subgraph to denote an induced subgraph unless stated otherwise.

2.2 Canonical Representation of Graphs

We represent every graph G by an adjacency matrix M. Slightly different from the adjacency matrix used for an unlabeled graph (Cormen et al, 2001), every diagonal entry of M represents a node in G and is filled with the label of the node. Every off-diagonal entry corresponds to a pair of nodes, and is filled with the edge label if there is an edge between these two nodes in G, or is zero if there is no edge.

Given an $n \times n$ adjacency matrix M of a graph with n nodes, we define the *code* of M, denoted by code(M), as the sequence of lower triangular entries of M (including the diagonal entries) in the order: $M_{1,1} M_{2,1} M_{2,2} \ldots M_{n,1} M_{n,2} \ldots M_{n,n-1} M_{n,n}$ where $M_{i,j}$ represents the entry at the *i*th row and *j*th column in M.

The standard lexicographic ordering of sequence defines a total order of codes. For example, code "ayb" is greater than code "byb" since the first symbol in string "ayb" is greater than the first symbol in string "byb" (We use the order a > b > c > d > x > y > 0). For a graph G, we define the Canonical Adjacency Matrix (CAM) of G as the adjacency matrix that produces the maximal code among all adjacency matrices of G. Interested readers might verify that the adjacency matrix M_1 in Figure 2 is the CAM of the graph P shown in Figure 1.

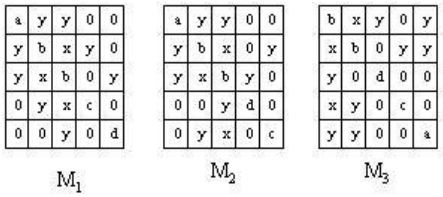


Figure 2. Three examples of adjacency matrices. After applying the total ordering, we have *code*(*M*1) = "*aybyxb*0*yxc*00*y*0*d*" > *code*(*M*2) = "*aybyxb*00*yd*0*yx*0*c*" > *code*(*M*3)

Given an $n \times n$ matrix N and an $m \times m$ matrix M, we define N as the *maximal proper submatrix (MP submatrix for short)* of M iff n = m-1 and $n_{i,j} = m_{i,j}$ $(0 < i, j \leq n)$.

One of the nice properties of the canonical form we are using (as compared to the one used in Inokuchi et al. 2000 and Kuramochi et al. 2001) is that, given a graph database GD, all the frequent subgraphs (represented by their CAMs) could be organized as a rooted tree. This tree is referred to as the *CAM Tree* of G and is formally described as follows:

- The root of the tree is the empty matrix;
- Each node in the tree is a distinct frequent connected subgraph of G, represented by its CAM;

- For a given none-root node (with CAM M), its parent is the graph represented by the MP submatrix of M;

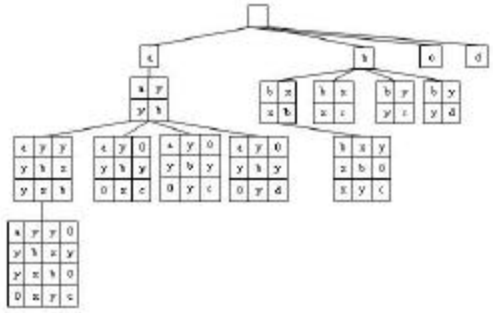


Figure 3. Tree organization of all the frequent subgraphs of the graph database shown in Figure 1 (a)

2.3 Finding Patterns from Labeled Graph Database

As mentioned earlier, the subgraph mining of protein databases presents a significant challenge because protein graphs are large and dense resulting in an overwhelmingly large number of possible subgraphs (Huan et al. 03). In order to select important features from the huge list of subgraphs, we have proposed a subgraph mining model based on mutual information as explained below.

2.3.1 Mutual Information and Coherent Induced Subgraphs

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

X_G = 1 with probability sup_G
0 with probability 1-sup_G

Given a graph G and its subgraph G', we define the mutual information I(G, G') as follows:
$I(G, G') = \sum_{X_G, X_{G'}} p(X_G, X_{G'}) \log_2(p(X_G, X_{G'})/(p(X_G)p(X_{G'})))$. where $p(X_G, X_{G'})$ is the (empirical) joint probability distribution of $(X_G, X_{G'})$, which is defined as follows:

$p(X_G, X_{G'}) =$	sup_G	if $X_G = 1$ and $X_{G'} = 1$
	0	if $X_G = 1$ and $X_{G'} = 0$
	$sup_{G'} - sup_G$	if $X_G = 0$ and $X_{G'} = 1$
	1- $sup_{G'}$	otherwise

Given a threshold t (t > 0) and a positive integer k, a graph G is *k-coherent* iff $\forall G' \subseteq G$ and $|G'| \geq k$, $(I(G, G') \geq t)$, where |G'| denotes the number of nodes in G'.

The *Coherent Subgraph Mining* problem is to find all the k-coherent subgraphs in a graph database, given a mutual information threshold t ($t > 0$) and a positive integer k.

Our algorithm for mining coherent subgraphs relies on the following two well-known properties (Tan et al. 2002):

Theorem For graphs $P \subseteq Q \subseteq G$, we have the following inequalities:

$$I(P, G) \leq I(P, Q)$$
$$I(P, G) \leq I(Q, G)$$

The first inequality implies that every subgraph G' (with size $\geq k$) of a k-coherent graph is itself k-coherent. This property enables us to integrate the k-coherent subgraph into any tree-based subgraph using available enumeration techniques (Yan and Han 2002, Huan et al. 2003). The second inequality suggests that, in order to tell whether a graph G is k-coherent or not, we only need to check all k-node subgraphs of G. This simplifies the search.

In the following section, we discuss how to enumerate all connected induced subgraphs from a graph database. This work is based on the algebraic graphical framework (Huan et al. 2003) of enumerating all subgraphs (not just induced subgraphs) from a graph database.

2.3.2 Coherent Subgraph Mining Algorithm

CSM
input: a graph database GD, a mutual information threshold t ($0 < t \leq 1$) and a positive integer k
output: set S of all G's coherent induced subgraphs.
P ← {all coherent subgraphs with size k in GD}
S ← F
CSM-Explore (P, S, t, k);

CSM-Explore
input: a CAM list P, a mutual information threshold t ($0 < t \leq 1$),
a positive integer k, and a set of coherent connected subgraphs' CAMs S.
output: set S containing the CAMs of all coherent subgraphs searched so far

For each $X \in P$
  $S \leftarrow S \cup \{ X \}$
  $C \leftarrow \{Y \mid Y$ is a CAM and X is the MP submatrix of $Y\}$
  remove non k-coherent element(s) from C.
  CSM-Explore(C, S, t, k)
End

3 Experimental Study

3.1 Implementation and Test Platform

The coherent subgraph mining algorithm is implemented using the C++ programming language and compiled using g++ with O3 optimization. The tests are performed using a single processor of a 2.0GHz Pentium PC with 2GB memory, running RedHat Linux 7.3. We used Libsvm for protein family classification (further discussed in Section 3.4); the Libsvm executable was downloaded from http://www.csie.ntu.edu.tw/~cjlin/libsvm/ .

3.2 Protein Representation as a Labeled Graph

We model a protein by an undirected graph in which each node corresponds to an amino acid residue in the protein with the residue type as the label of the node. We introduce a "peptide" edge between two residues X and Y if there is a peptide bond between X and Y and a "proximity" edge if the distance between the two associated C_α atoms of X and Y is below a certain threshold (10Å in our study) and there is no peptide bond between X and Y.[1]

3.3Datasets and Coherent Subgraph Mining

Three protein families from the SCOP database (Murzin et al, 1995) were used to evaluate the performance of the proposed algorithm under a binary (pair-wise) classification scheme. SCOP is a domain expert maintained database, which hierarchically classifies proteins by five levels: Class, Fold, Superfamily, Family and individual proteins. The SCOP families included the Nuclear receptor ligand-binding domain (NRLB) family from the all alpha proteins class, the Prokaryotic serine protease (PSP) family from the all beta proteins class, and Eukaryotic serine protease (ESP) family from the same class. Three datasets for the pairwise comparison and classification of the above families were then constructed: C1, including NRLB and PSP families; C2, including ESP and PSP families, and C3, including both eukaryotic and prokaryotic serine proteases (SP) and a random selection of 50 unrelated proteins (RP). All the proteins were selected from the culled PDB list, (http://www.fccc.edu/research/labs/dunbrack/pisces/culledpdb.html) with less than 60% sequence homology (resolution = 3.0, R factor = 1.0) in order to remove redundant sequences from the datasets. These three datasets are further summarized in Table 1.

For each of the datasets, we ran the coherent subgraph identification algorithm. Thresholds ranging from 0.5 to 0.25 were tested; however, we only report the results with threshold 0.3, which gave the best classification accuracy in our experiments.

---

[1] Note that this graph representation provides a lot of flexibility for future studies, e.g. using smaller number of residue classes or using additional edge labels.

3.4 Pair-wise Protein Classification Using Support Vector Machines (SVM)

Given a total of n coherent subgraphs $f_1, f_2, \ldots, f_n$, we represent each protein G in a dataset as a n-element vector $V=v_1, v_2, \ldots v_n$ in the feature space where v_i is the total number of distinct occurrences of the subgraph f_i in G (zero if not present). We build the classification models using the SVM method (Vapnik 1998). 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 only occur in a small set of samples), 2) there are a set of kernel learning functions (such as linear, polynomial and radius based) we could choose from, depending on the property of the dataset.

Table 1 summarizes the results of the three classification experiments and the average five fold cross validation total classification accuracy [i.e., (TP + TN)/(N) where TP stands for true positive, TN stands for true negative, and N is the total number of testing samples]. In order to address the problem of possible over-fitting in the training phase, we created artificial datasets with exactly same attributes but randomly permuted class labels. This is typically referred to as the Y-randomization test. The classification accuracy for randomized datasets was significantly lower than for the original datasets (data not shown) and hence we concluded that there is no evidence of over-fitting in our models.

	Class A	Total # Proteins	Class B	Total # Proteins	Features	Time, (sec.)	Accuracy (%)
C1	PSP	9	NRLB	13	40274	240	96
C2	PSP	9	ESP	35	34697	450	93
C3	SP	44	RP	50	42265	872	95

Table 1. Accuracy of classification tasks C_1, C_2, C_3. We used the C-SVM classification model with the linear kernel and left other values as default. Columns 1-4 give basic information about the dataset. SP –serine proteases; PSP – prokaryotic SP; ESP – eukaryotic SP; NRLB – nuclear receptor ligand binding proteins, RP – random proteins. The fifth column (Features) records the total number of features mined by CSM and the sixth column (Time) records how much CPU time was spent on the mining task. The last column gives the five fold cross validation accuracy.

3.5 Identification of Fingerprints for the Serine Protease Family

Features found for the task C3 in Table 1 were analyzed to test the ability of the CSM method to identify recurrent sequence-structure motifs common to particular protein families; we used serine proteases as a test case. For every coherent subgraph, we can easily define an underlying elementary sequence motif similar to Prosite patterns as:

$M = \{AAp, d_1, AAq, d_2, AAr, d_3, AAs\}$

where AA is the residue type, p, q, r and s are residue numbers in a protein sequence, and d_1=q-p-1, d_2=r-q-1, d_3=s-r-1, i.e., sequence separation distances.

We have selected a subset of the discriminative features from the mined features such that every feature occurs in at least 80% of the proteins in the SP family and in less than 10% of the proteins of the RP family. For each occurrence of such features, sequence distances were analyzed. Features with conserved sequence separation were used to generate consensus sequence motifs. We found that some of our spatial motifs correspond to serine protease sequence signatures from the Prosite Database. An example (G1) of such a spatial motif and its corresponding sequence motif C-x(12)-A-x-H-C (where x is any residue(-s) and the number in the parenthesis is the length of the sequence separation) are shown in Fig. 4. This example demonstrates that the spatial motifs found by subgraph mining can capture features that correspond to motifs with known utility in identifying protein families. The spatial motif G2, which also was highly discriminative, occurs in SP proteins at a variety of positions, with varying separations between the residues. Such patterns seem to defy a sequence-level description, hence raise the possibility that spatial motifs can capture features beyond those described at the sequence level.

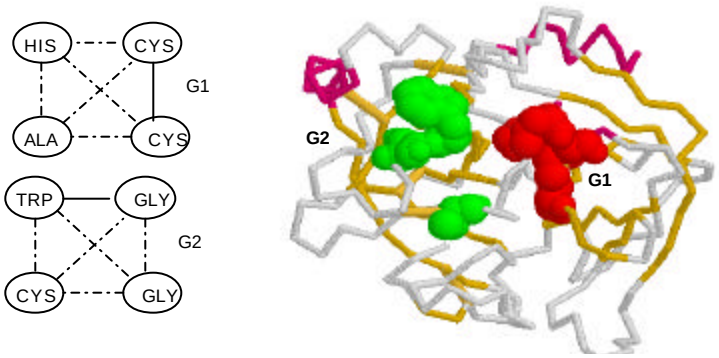


Figure 4: Two discriminative features that appear very frequently in SP family while are infrequent in the RP family. Left: the graphical representation of the two subgraphs (with residue type specified within the circle). A dotted line in the figure represents a proximity edge and a solid line represents a peptide edge. Right: the 3D occurrences of G1 (right) and G2 (left) within the backbone of one of serine proteases, Human Kallikrein 6 (Hk6).

4 Conclusions and Future Work

We have developed a novel coherent subgraph mining approach and applied it to the problem of protein structural annotation and classification. As a proof of concept, characteristic subgraphs have been identified for three protein families from the SCOP database, i.e., eukaryotic and prokaryotic serine proteases and nuclear receptor binding proteins. Using Support Vector Machine binary classification algorithm, we have demonstrated that coherent subgraphs can serve as unique structural family identifiers that discriminate one family from another with high accuracy. We have also shown that some of the subgraphs can be transformed into sequence patterns similar to Prosite motifs allowing their use in

the annotation of protein sequences. The coherent subgraph mining method advanced in this paper affords a novel automated approach to protein structural classification and annotation including possible annotation of orphan protein structures and sequences resulting from genome sequencing projects. We are currently expanding our research to include all protein structural families and employ multi-family classification algorithms to afford global classification of the entire protein databank.

Acknowledgments

The authors would like to thank Prof. Jack Snoeyink and Deepak Bandyopadhyay for many helpful discussions.

References

1. R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *Proc. of the 20th Int. Conf. on Very Large Databases (VLDB)*, 487–499 (1994)
2. T. Akutsu, H. Tamaki 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*, 314-323 (1997)
3. S. Chakraborty and S. Biswas, "Approximation Algorithms for 3-D Common Substructure Identification in Drug and Protein Molecules", *Workshop on Algorithms and Data Structures*, 253-264 (1999)
4. T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, (MIT press, 2001).
5. 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*, 30-6 (1998)
6. M. Deshpande, M. Kuramochi and G. Karypis, "Frequent Sub-Structure-Based Approaches for Classifying Chemical Compounds", *Proc. of the 8th International Conference on Knowledge Discovery and Data Mining* (2002)
7. 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*, 324-333 (1997)
8. 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**, 769-778 (1994)

9. S Henikoff, J Henikoff, S Pietrokovski. “Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations”, *Bioinformatics*, **15**(6):471-9 (1999)
10. K. Hofmann, P. Bucher, L. Falquet, A. Bairoch, “The PROSITE database, its status in 1999”. Nucleic Acids Res, 1;27(1):215-9 (1999)
11. L. B. Holder, D. J. Cook and S. Djoko, “Substructures discovery in the subdue system”, Proc. AAAI'94 Workshop Knowledge Discovery in Databases, 169-180 (1994).
12. J. Huan, W. Wang, J, Prins, “Efficient Mining of Frequent Subgraph in the Presence of Isomorphism”, *Proc. of the 3rd International conference on Data Mining*, (2003)
13. P. Indyk, R. Motwani, S. Venkatasubramanian, Geometric Matching Under Noise, “Combinatorial Bounds and Algorithms”, *ACM Symposium on Discrete Algorithms* (1999).
14. A. Inokuchi, T. Washio, and H. Motoda, “An Apriori based algorithm for mining frequent substructures from graph data”, *In Proc. of the 4th European Conf. On Principles and Practices of Knowledge Discovery in Databases*, 13–23 (2000).
15. G.J. Kleywegt “Recognition of spatial motifs in protein structures” *J Mol Biol*. **285**(4):1887-97 (1999)
16. M. Kuramochi and G. Karypis, “Frequent subgraph discovery”, *Proc. of the 1st International conference on Data Mining*, (2001)
17. AG Murzin, SE Brenner, T Hubbard, C Chothia, “SCOP: a structural classification of proteins database for the investigation of sequences and structures”, *J. Mol. Biol*. **247**, 536-540 (1995)
18. CA Orengo and WR Taylor, “SSAP: Sequential Structure Alignment Program for Protein Structure Comparison”, *Methods in Enzymol* **266**: 617-643 (1996)
19. P. Tan and V. Kumar and J. Srivastava, "Selecting the right interestingness measure for association patterns", *In Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining* (2002)
20. V. Vapnik, *Statistical Learning Theory*, (John Wiley, 1998)
21. X. Yan and J. Han. gSpan, “Graph-based substructure pattern mining”, *Proc. of the 2nd International conference on Data Mining*, (2002)
22. *K. Yoshida and H. Motoda*, “CLIP: Concept learning from inference patterns”, *Artificial Intelligence*, **75**(1):63–92, (1995)