International Journal of Research
Available at https://edupediapublications.org/journals
Special Issue on Conference Papers
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 06
March 2018

Frequent and Significant Patterns Mining Using Approximate Patterns for Protein Structure Analysis

[1]D.KAVITHA, [2] V. KAMAKSHI PRASAD, [3]J.V.R. MURTHY
[1]SrAssistant Professor, PVPSIT , Vijayawada
Professor & Director of Evaluation, JNTUH College of Engineering, JNTUH
Professor, Dept of Computer Science, J.N.T.U Kakinada, INDIA
[1]dkavithad@gmail.com, [2]kamakshiprasad@jntuh.ac.in, [3]mjonnalagedda@gmail.com

Abstract: With the advent of technology and tools, large volumes of data have generated in varying complex forms. Graphs and graph based data mining has emerged as an appropriate solution to represent and to mine useful knowledge from such complex structured data. In order to extract useful information from these datasets, many algorithms are being developed for various graph based data mining tasks such as frequent pattern mining, classification, clustering and indexing in graph data. Graphs are especially appropriate to model proteins structures and to determine the structural and functional characteristics of different proteins which is evolving as a key area in genetic information processing. In this paper, we present FSPM, a novel framework which retrieves both frequent and significant patterns directly at a time using approximate patterns. To the best of our knowledge this is the first work that mines frequent and significant patterns at a stretch. Our preliminary experiments demonstrate the efficiency of the proposed framework.

I. INTRODUCTION:

Graphs are being progressively more used to model a broad range of scientific data such as graphical symbol recognition, shape analysis, protein structure analysis, computer network monitoring, web data analysis, social networks, XML[1-4] data and so on. Such widespread usage of graphs has generated substantial interest in mining patterns from graph databases[6-8]. Mining graph patterns will facilitate to understand the inherent characteristics and behaviour.

Further particularly in biology[5], graphs are especially appropriate to model proteins structures and interactions between different proteins. Studying protein structures can disclose appropriate structural and functional information which may not be derived from protein sequences alone. In this context, recently, proteins have been interpreted as graphs of amino acids and studied based on graph theory concepts [5]. These representations permit the use of graph mining techniques to study protein structures in a graph perception. In fact, in graph mining, any application under consideration is represented in the form of nodes and edges and solved based on graph theory concepts. Yet, the exponential growth of online databases such as the Protein Data Bank (PDB) [19], CATH[27] , SCOP and others, arises an urgent need for more accurate methods that will help to better understand the studied phenomenon such as protein evolution, functions, etc.

Protein Structure is described in four levels: (a) The primary structure is the succession of amino acid residues, usually abbreviated by the 1- or 3-letter codes.
(b) The secondary structure is the 3-D arrangement of the right-handed alpha helix (shown here), or alternative structures such as a beta-pleated sheet.
(c) The tertiary structure of a protein is the 3-D folding of the alpha helix (a purple ribbon), shaped by structures such as proline corners, disulfide bridges between cysteine residues, and electrostatic bonds.
(d) Where more than one protein chain contributes to the protein, the quaternary structure is the arrangement of these

subunits. In haemoglobin as shown here, the quaternary structure comprises two alpha and two beta polypeptides, held together by electrostatic bonds.[23]

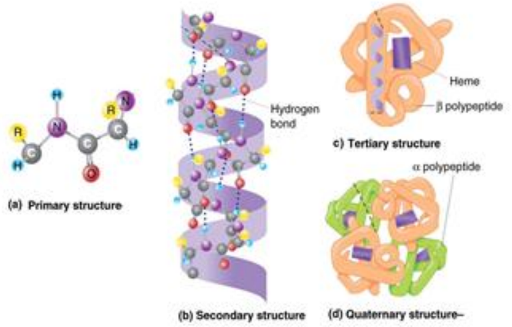


Fig. 1 Protein Structure described in four levels

One of the powerful and current trends in graph mining is frequent subgraph discovery. It aims to discover subgraphs that frequently occur in a graph dataset and use them as patterns to describe the data. These patterns are lately analyzed by domain experts to reveal interesting information hidden in the original graphs, such as discovering pathways in metabolic networks, identifying residues that play the role of hubs in the protein and stabilize its structure etc.

An amino acid basic unit of protein, consists of a central carbon atom attached to a carboxyl group (-COO), an amino group (-NH$_3$), a hydrogen atom and a side group (-R), giving the general formula R-CH-NH$_3$-COO shown in figure 2. Only the side group differs from one amino acid to another. The side chain determines the characteristic properties[24] of each amino acid and the side-chain groups vary in size, shape, charge etc. Carboxylic group and amino group are present in each of the amino acid and are useful to identify the group characteristics of amino acids. By identifying the frequent subgraphs present in set of those protein graphs, we can conclude that amino acids contain those groups and they differ by side chain. Thus frequent subgraph mining is useful to characterize and to identify basic appearances of amino acids.

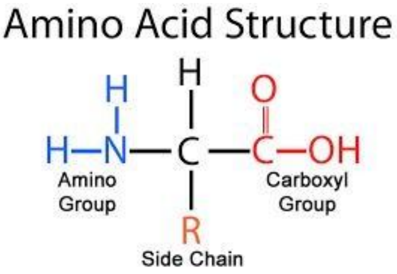


Fig. 2 Frequent Pattern in Amino Acids

On the other hand as shown in Figure3, Asp, Glu,Lysine and arginine are classified as electrically charged which are further classified as Acidic and basic, and Ser, Thr, asn and Gln are classified as uncharged polar group based on the differences in the side chain R. i.e., we have to identify how side chain differs in amino acids which lead to the classification of amino acids into negatively charged R groups, positively charged R groups, aromatic R groups etc. But for the classification, the substructures identified in frequent mining isn't enough. We have to retrieve the substructures that have significance to make classification among those which is leaded by significant subgraph mining. So frequent patterns discovered by frequent subgraph mining algorithms are useful for general characterization or for broad classification of groups where as in particular, significant subgraph mining, variant of frequent subgraph mining aims at mining significant patterns to discriminate and to identify significant characteristics of amino acids. Based on this example, we have chosen the problem 1) identification of frequent subgraphs in an efficient and optimized manner, 2) identification of significant subgraphs framework

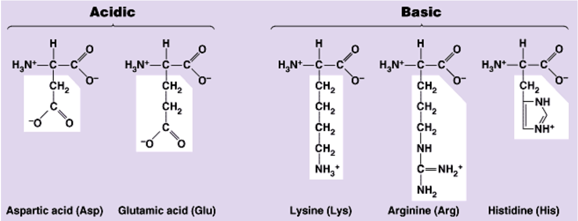


Figure3: Classification of some Amino Acids based on inherent information

Redundancy in a frequent subgraph set is caused by structural and/or semantic similarity, since most discovered subgraphs differ slightly in structure and may infer similar or even the same meaning. Moreover, the significance of the discovered frequent subgraphs is only related to frequency. This yields an urgent need for efficient approaches allowing to select relevant patterns among the large set of frequent subgraphs.

Challenges:

Frequent subgraph mining:

Frequent subgraph mining is the process of discovering frequent subgraphs from graph data. It is one of the most powerful and challenging task in graph mining. In general, frequent subgraph discovery consists of the following steps 1) candidate generation 2) candidate pruning and 3) support counting. In the candidate generation step, frequent subgraph candidates are generated, candidate pruning is the process of discarding candidate k-edge subgraphs that have infrequent k-1 edge subgraphs and support counting step is carried out to check the frequency of candidate subgraph. The main challenge lies in frequent pattern mining is explosive generation of subgraphs in the process of candidate generation.

Significant Subgraph Mining:

Generally in significant pattern mining process, at first, all frequent subgraphs are mined and then significant patterns are selected from them based on user defined objective function. Clearly, this two-step traditional procedure is not scalable to find significant subgraphs due to the reasons -- low frequency threshold has to be set for many objective functions in significant pattern mining which generates an exponential search space and slow mining process. Hence, mining frequent patterns with low threshold becomes a challenge of the mining process.

The remainder of the paper is organized as follows. Section 2 discusses the recent related works in the area of pattern selection for subgraphs. In Section 3, we present the background of our work and we define the preliminary concepts as well as the main algorithm of our approach. Then, Section 4 presents the obtained experimental results and the discussion and Section 5 presents the conclusion.

II. RELATEDWORK

Here are some existing algorithms that mine frequent and significant patterns and find use in applications. There have been popular graph mining algorithms based on graph theory are proposed such as AGM (Apriori-based Graph Mining) [8], FSG (Frequent SubGraph discovery) [9], gSpan (graph-based Substructure pattern mining) [10], and FFSM (Fast Frequent Subgraph Mining) [12]. gSpan and FSG needs a lot of time to solve subgraph isomorphism problem. FFSM converts subgraph isomorphism problem into graph isomorphism problem, but testing of graph isomorphism still need lot of time. Ranu and Singh [13] used a feature vector representation to find significant patterns. Unfortunately, this method need to mine all of (closed) frequent subgraphs first. Yan et al[14] proposed leap search to find patterns, focussed on the databases that can be divided into positive and negative sets. Deshpande et al. [15] used frequent structures as features to classify chemical compounds and Yan et al. [25] used as indexing features to perform fast graph search. Pattern-based classification models were demonstrated in [27] which use only significant discriminative patterns, where complete sets of frequent subgraphs could even bring poor performance and low accuracy, e.g., redundant indices and over fitted classifiers. Hasan et al. [16] presented mining a set of representative orthogonal patterns using a randomized search approach. Kudo et al. [18] presented an application of boosting for classifying labeled graphs, such as chemical compounds, natural language texts, *etc.*

We propose a framework that addresses the above mentioned challenges in mining frequent graph patterns which also exploit the significant patterns. The proposed algorithm is able to find frequent and significant patterns in finite amount of time and in a scalable manner by using a small set of approximate patterns.

III. Pattern Mining using Approximations

In this paper, we propose a novel pattern retrieval approach which selects a subset of approximate patterns from a set of labeled patterns. Moreover, this approach is unsupervised and can help in various mining tasks, unlike other approaches that are intended and dedicated to a specific task such as classification. In order to select these approximate patterns, we exploit a specific domain knowledge, which is the substitution between amino acids represented as nodes. Though, the main contribution of this work is to define a new approach for mining a representative summary of the set of patterns using different categories of frequent patterns that are helpful to identify the general and specific characteristics. In this work, we apply the proposed approach on protein structures because of the availability of structures in the literature, however, it can be considered as general framework for other applications whenever it is possible to define approximate patterns. Our approach can also be used on any type of subgraph structure such as cliques, trees and paths (sequences). In addition, it can be easily coupled with other pattern selection methods such as discrimination or orthogonality based approaches. Moreover, this approach is unsupervised and can help in various mining tasks, unlike other approaches that are supervised and dedicated to a specific task such as classification.

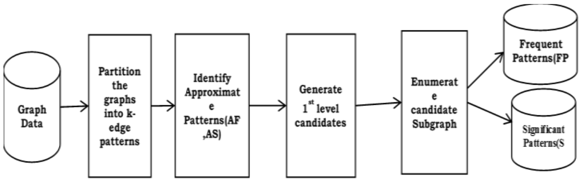

Graph Data
Partition the graphs into k-edge patterns
Identify Approximate Patterns(AF ,AS)
Generate 1st level candidates
Enumerate candidate Subgraph
Frequent Patterns(FP
Significant Patterns(S

Fig.4: Overview of the Proposed Framework

In summary, our contributions are:

- A novel frame work to mine frequent and significant patterns in a data set is proposed. It is based on an idea to avoid enumeration of all frequent patterns which cause redundancy and extreme mining time.
- It offers a heuristic search with novel optimizations to significantly improve performance by pruning the search space by exploring only approximate patterns.
- It offers to mine frequent patterns and significant patterns directly at a time.

Preliminaries:

Definition1: Pattern: Let S be a set of n labeled patterns where S = {P_1, $P_{2,\ldots,}P_n$}. Each labeled pattern is represented with P= (V, E, L) where V is a finite set of vertices, E is a set of edges and E$\subset$V $\times$V, L is a set of labels. We assume that the pattern P is connected and undirected that is each edge is an unordered pair of vertices. Furthermore the pattern is labeled that is each vertex and edge has a label associated with it from defined set of labels L. Each vertex v(or edge e) of the pattern P is not required to have a unique label and the same label can be assigned to many vertices (or edges) in the same graph. If all the vertices and edges of the graph have the same vertex and edge label assigned to them, we will call this graph unlabelled.

Definition2: Subpattern: A pattern p=(V', E', L') is a subpattern of another pattern P=(V, E, L) iff V'$\subseteq$ V, and E'$\subseteq$ E $\wedge$ ((v_1, v_2)$\in$ E' $\rightarrow$ v_1, $v_2$$\in$ V') and it holds that (lbl(u) = lbl' (μ(u))) $\wedge$ (lbl(v) = lbl'(μ(v))) $\wedge$ (lbl(u, v) = lbl'(μ(u), μ(v))). In other words, the labels for each corresponding edge as well as the labels of edge's end points i.e., labels of vertices are to be identical. p $\subseteq$ P notifies that p is a subpattern of pattern P.

Definition3: Isomorphism: For two labelled patterns P_1 and P_2, an isomorphism is a bijection f: V(P_1) $\rightarrow$ V(P_2) such that $\forall$ v $\in$ V(P_1), L(v) = L'(f (v)) and $\forall$ (u,v) $\in$ E(P_1)$\Leftrightarrow$ (f(u), f(v)) $\in$ E(P_2) and L(u, v) = L'(f (u), f (v)) where L and L' are labels of P_1 and P_2 respectively. This mapping preserves labels on the vertices and edges.

Definition4: Frequency: The frequency of a pattern p is defined as number of graphs in S that contains p as a subpattern. Given a data set S = {P_1, $P_{2,\ldots.,}P_n$} and a subpattern p, the frequency f of p, is

$$f = \frac{|Pi|}{|S|}.$$

Definition5: Frequent Pattern: Given a dataset S= {P_1, P_2,…,P_n}, a frequency threshold T and a pattern p with frequency f, then a pattern p is frequent if and only if $f \geq T$ w.r.t |S|

Definition6: Significant Pattern: Given a data set S= {P_1,P_2,…P_n} and an objective function F, a general problem formulation for mining significant patterns can be of two ways:

(1) find all patterns p* such that $F(p) \geq T$ where T is a significance threshold -- that is significant patterns p*, the set of patterns having an objective score greater than or equal to threshold and objective function F is a threshold based function;

(2) Find a pattern p* such that p* = $\text{argmax}_p F(p)$ -- that is p* the set of patterns that maximizes the objective function F.

Definition7: Approximate Pattern Set: An approximate pattern set is a pattern subset *A* with respect to the given graph dataset *S* that approximates the original patterns and result, that is finding the approximate pattern for the given objective function *F* provides an approximate solution for the problem on the dataset *S*

Algorithm:

Given a set of patterns, we propose mining using approximation, a pattern selection algorithm which selects set of frequent patterns and sets of discriminative patterns. Based on our approximation concept, all the patterns in both categories are identified at a time.

The general process of algorithm is described as follows- First the graph is divided into sub patterns having the same number of edges. Then each pattern is canonically ordered. Then identify subsets of a graph by identifying the repeated patterns in a graph. This process is repeated for all graphs in a data set. Now, each subset represents a pattern .Next identify the frequency of each pattern in a whole data set. Each pattern represents an approximation of result. Normalize the resulted frequencies and identify the thresholds for frequent subgraphs and substantial patterns. Extend the approximated patterns which are above threshold in their respective categories and finally identify patterns that belong to those categories with in a same framework.

Algorithm: Frequent and Significant Pattern Mining(FSPM)

Input: A graph dataset *D*

Output: frequent patterns *FP* and significant patterns *SP*

```
begin

    $F^1 \leftarrow$ All frequent 1- edge subgraph in D in Canonical edge form
    findapproximateset ($F^1$)
    for each $s$ belongs to $AS/AF$ do
       $FP,FS \leftarrow 0$
    while $AS/AF \neq \emptyset$ do
       for each $s$ in $AS/AF$ do
            let e' be last edge of $s$ and for each edge $e$ in $F^1$
            if $e$ can be used to extend $e'$ then
            $ext \leftarrow s <> e$
            if $ext$ is not already generated then
            $AS^j \leftarrow AS^j \cup ext$ ( or) $AF^j \leftarrow AF^j \cup ext$
         For each c in $AF$
            if c.$fcnt \geq t$ or then
              $FP^k \leftarrow FP^k \cup c$
        For each c in $AS$
            if c.$fcnt \geq t$ or then
              $SP^k \leftarrow SP^k \cup c$

end
```

Algorithm: *findapproximateset()*

Input: $F^1 \leftarrow$ All frequent 1- edge subgraph in D in Canonical edge form

Output: Approximate Patterns *AS, AF*

$S^k \leftarrow$ All k- edge patterns in D in Canonical edge form

AS, AF $\leftarrow$ data structures to store Approximate patterns

Begin

```
$S^k$ ← k-edge patterns generated from $F^1$
$S$← $S^k$  All subgraphs with k-edge in edge-format
for each $s$ in $S$ do
    find $fcnt$ of $s$ in $S$
find $t$ and $T$   // set threshold $T$ and $t$ – $min$  and $max$ $threshold$  based on ranges appeared and user given significance
for each $s$ in $S$ do
    if  $s.fcnt \geq t$ or then
        $AF^k \leftarrow AF^k \cup s$
    if  $s.fcnt \geq T$ then
        $AS^k \leftarrow AS^k \cup s$
end
```

IV. Experimental Evaluation

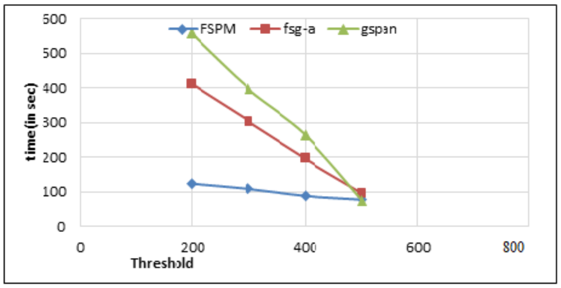
FSPM Algorithm is implemented in C language. The experiments are carried out on a Intel® Pentium® Dual CPU T3400 @2.17 GHz with 4GB RAM. To evaluate the performance on real datasets, we used the data sets in a standard graph library available at the Pubchem. (1)Pubchem[22] is a well maintained compilation of various molecules. It consists of 1178 proteins, which can again be divided up into two classes: 691 enzymes and 487 non-enzymes. Average vertices per graph are 285 and edges are 715. Different vertex labels available are 82.

(2)Another protein data set is from Protein Data Bank[19]. Each Protein is represented with a graph of amino acids. Each node in a protein represents an amino acid and is labeled with its label. Average vertices per graph are 189 and edges are 745. Different vertex labels available are 92and maximum vertices and edges are 755and 3012.

(3)Another protein data set is from [20] D&D benchmark. This is a DIP database contains proteins as nodes. This data set contains 1178 proteins that are divided into 691 enzymes and 487 non-enzymes. Average vertices per graph are 285 and edges are 715. Different vertex labels available are 82.

The scalability of the algorithm against frequency and data set size are examined. For these experiments FSGand gSpan comprised with significant mining are chosen to compare. The executable programs for both FSG and gSpan algorithm was obtained from the home page of respective authors, Karypis[9], and Xeifeng Han[21]. The considered datasets are converted into acceptable format of these programs and the executed results of those algorithms are pipelined to acquire significant patterns for frequent and significant pattern mining.

Figure 5 shows the performance of algorithms with respect to frequency threshold and time for different data sets. In general the execution time required to generate frequent subgraphs at low threshold is exponential when compare with the time required at high threshold. This is because of the number of frequent subgraphs growing exponentially as the threshold decreases. So that the required time to execute also increases. The algorithm shows linear increase in execution time against threshold as it identifies the frequency at the time of partitioning. For other algorithms, the rate of increase in runtime is exponential when the frequency threshold decreases. Unlike other algorithms, the proposed algorithm does not need to generate all intermediate subgraphs. Consequently, the computation required to perform isomorphism testing for all intermediates is pruned, thus, it is efficient. In third one, performance using gspan is unavailable as it is inefficient if number of vertices are greater than 250 vertices.



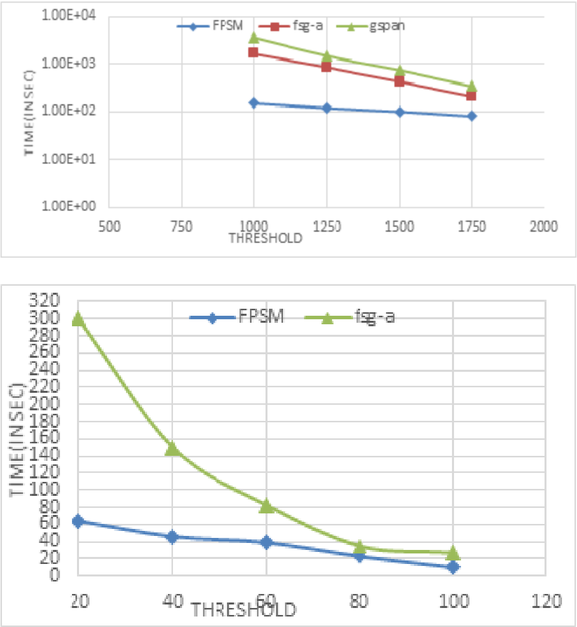


Figure 5: The efficiency of Algorithm in mining frequent Patterns

Figure 6 shows the scalability of the proposed algorithm with respect to frequency threshold for mining significant subgraphs. The frequency threshold varied between 1% and 10%. From the results shown in figure, it can be inferred that the time taken to find the significant patterns for the proposed mining approach is less when compared with the other two approaches. While comparing the performance at low frequency threshold, proposed algorithm exhibited linear behaviour and the other two are exponential in nature. But, at higher frequency the gSpan is little bit faster than the proposed one as the proposed algorithm requires computation of coreset where as those start mining directly. However, it is insignificant when compared with the cost of mining at less frequency thresholds.

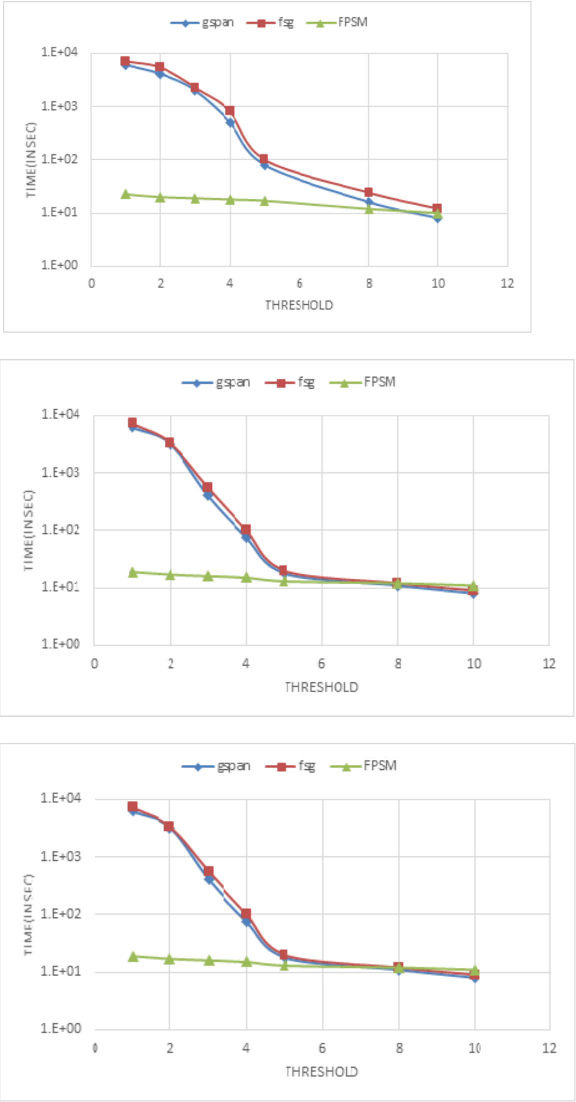


Figure 6: Performance of algorithm to mine significant patterns

V. CONCLUSION:

In this work, the emphasis of proposed framework is to provide a scalable approach to mine frequent as well as significant patterns at low frequency threshold from graph database thus make available to perform graph mining tasks like analysis, classification and indexing with accuracy and efficiency in a scalable manner. The proposed *approximate patterns* concept evaluates the frequency of the subgraphs based on topological structure, frequency and similarity between candidate subgraphs. The proposed technique reduces exponential search space and determines representative patterns based on the frequency. As a result, candidate generation can be performed with a drastic reduction in computational cost. The proposed approximate pattern based mining technique is capable of retrieving the required patterns directly from the graphs consequently avoiding unnecessary subgraph extensions and reducing computation cost. The proposed framework is capable of retrieving patterns of different frequencies at a time. The results obtained in the preliminary tests confirmed the effectiveness of the proposed algorithm. The proposed algorithm can also be applied to search structures based on the user given constraints for general applications.

VI. References:

[1] R. Kumar, P Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, E. Upfal. The Web as a Graph. ACM PODS Conference, 2000.

[2] D. Maio and D. Maltoni. A structural approach to fingerprint classification. In Proceedings of 13th International Conference on Pattern Recognition, Vienna, Austria, pp. 578–585, 1996

[3] F. Eichinger, K. Bohm, M. Huber. Improved Software Fault Detection - with Graph Mining. Workshop on Mining and Learning with Graphs, 2008

[4] M. S. Gupta, A. Pathak, S. Chakrabarti. Fast algorithms for top-k personalized pagerank queries. WWW Conference, 2008

[5] M. Koyuturk, A. Grama, W. Szpankowski. An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks. Bioinformatics, 20:I200–207, 2004.

[6] B. Zhou, J. Pei. Preserving Privacy in Social Networks Against Neighborhood Attacks. ICDE Conference, pp. 506-515, 2008

[7] A. Inokuchi, T. Washio, and H. Motoda, “An apriori-based algorithm for mining frequent substructures from graph data,” in Principles of Data Mining and Knowledge Discovery, 2000, pp. 13–23..

[8] S. Nijssen and J. N. Kok, “The Gaston tool for Frequent Subgraph Mining,” in Proceedings of the International Workshop on Graph-Based Tools, Grabats 2004, Rome, Italy, October 2, 2004. Elsevier, 2004.

[9] M. Kuramochi, G. Karypis, An efficient algorithm for discovering frequent subgraphs, IEEE Trans. Knowl. Data Eng. 16 (9) (2004) 1038–105.

[10] X. Yan, J. Han, gSpan: graph-based substructure pattern mining, Proceedings of 2002 IEEE Int'l Conference on Data Mining, 2002.

[11] A. Inokuchi, T. Washio, H. Motoda, Complete mining of frequent patterns from graphs: mining graph data, Mach. Learn. 50 (2003) 321–354.

[12] J. Huan, W. Wang, J. Prims, Efficient mining of frequent subgraph in the presence of isomorphism, University of North Carolina Computer Science Technical Report, 2003.

[13] Sayan Ranu , Ambuj K. Singh, GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases, In the proceedings IEEE International Conference on Data Engineering 2009

[14] X. Yan, H. Cheng, J. Han, and P. S. Yu, “Mining Significant Graph Patterns by Scalable Leap Search,” in Proceedings of SIGMOD, 2008.

[15] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent substructure-based approaches for classifying chemical compounds. IEEE Transactions on Knowledge and Data Engineering 17(18):1036–1050, 2005

[16] M. Hasan, V. Chaoji, S. Salem, J. Besson, and M. Zaki. ORIGAMI: Mining representative orthogonal graph patterns. In Proc. of ICDM, pages 153 -162, 2007.

[17] S. Kramer, L. D. Raedt, and C. Helma, “Molecular feature mining in HIV data,” in *KDD ’01: Proceedings of the*

seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, pp. 136–143

[18] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. In Advances in Neural Information Processing Systems 18 (NIPS'04), 2004

[19] H. M. Berman, J. D. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. Nucleic Acids Research, 28(1):235–242, 2000

[20] DIP;http://dip.doe-mbi.ucla.edu

[21] X.Yan - http://www.cs.ucsb.edu/~xyan/software/gSpan.htm

[22] http://pubchem.ncbi.nlm.nih.gov

[23] Figure after © 2010 PJ Russell, iGenetics 3rd ed.; all text material © 2014 by Steven M. Carr

[24] Part of figure, Campbell's *Biology, 5th Edition*

[25] Yan X, Yu PS, Han J. Graph indexing: A frequent structure-based approach, In Proceeding Of SIGMOD, NY, pp. 335-46,2004

[26] Cheng H, Yan X, Han J, Hsu C. Discriminative frequent pattern analysis for effective classification, In Proceeding of ICDE, Istanbul, pp 716-725,2007

[27] A. L. Cuff, I. Sillitoe, T. Lewis, A. B. Clegg, R. Rentzsch, N. Furnham, M. Pellegrini-Calace, D. T. Jones, J. M. Thornton, and C. A. Orengo Extending cath: increasing coverage of the protein structure universe and linking structure with function. Nucleic Acids Research, 39:420-426, 2011.