International Journal of Engineering & Technology, 7 (4.6) (2018) 230-234

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Finding Frequent Patterns in Biological Networks

Rahul Rane[1]* and Praveen Kaushik[2]

[1]Department of Computer Science, Maulana Azad National Institute of Technology, Bhopal, India
[2]Department of Computer Science, Maulana Azad National Institute of Technology, Bhopal, India
**Corresponding author E-mail: rahulrane165@gmail.com*

Abstract

Frequent patterns help to discover the structural behavior of the complex biological network. The networks which show same global structure may have different local structure. The patterns are the actual fingerprints of the network. The pattern frequency and its significance carries very important functionality to classify and cluster the biological network. The sheer number of patterns get generated while traversing the network. Counting these patterns and determining their frequencies in the large biological network is very challenging and computationally expensive task. Previously proposed methods are bound by the size of patterns and networks size. By using approximation method like sampling, we proposed an algorithm. The results show the proposed algorithm is faster than existing algorithms. Also, the error rate is minimum, which makes the proposed method more reliable.

Keywords: *MCMC; Network motifs; Sampling algorithms.*

1. Introduction

The biological network consists of protein, genes, and molecules. The logical interaction between them forms a network in which every node act as a protein or genes or molecules and edges are the relationship between them. The interaction shows they share certain properties if they are connected. With the help of graph theory, it is easy for biological network data to revile the functionality of the network. By modeling biological data into graphs helps to predict protein-protein interaction [1], prediction of biological function, finding hidden interaction, drug designing [2].

The motifs called to be the pattern overly frequent and have statistical significance compared to the random graph network. Motif reveal key functionality of the biological network. The global properties of biological structure may have same, but there is a possibility of different local structure also. To find out the difference between the networks, motifs play the key role to them. The application of motifs in biology and medicals filed are studying genes and their functions, study interaction and analysis in protein, determining protein function [3] and finding essential genes and proteins [4].

Finding the frequent pattern in a Protein-Protein Interaction (PPI) network is currently major challenging task for computational biology because of growing size of networks. Even distribution of pattern topologies in interaction graph gives the edge to distinguish among the species. The PPI network has many species which shows same similar global structure but they have different local structure. Also some of the species follow some structure which is inherited from other species. It makes absolute need to find frequent pattern efficiently to study about different species.

The different notion has been proposed to generate the synthetic graph which has similar properties like a real-life biological graph. The progress concurs many benchmarks to generate exact structure but it gets very tough for the complex biological network. That makes hardly useful to counter real life biological network. Discovering the position and interaction of vertices in a pattern space which helps to solve different problems in the biological networks. Some of this problems are classifying structural difference, characterizing network using degree distribution, mapping functionality to the network. The pattern also acts as features for so many machine learning task to predict and infer important analysis.

The existing notions of graph patterns are Graphlet [5] and Motif [6]. The graphlets are embedded induced subgraph of different size as a part of the large graph. And motifs are the partial subgraph which may contain fewer edges in a large graph. In this paper, we mention the subgraph as a pattern.

Table 1: Number of distinct topologies

Vertex	Pattern
1	1
2	1
3	2
4	6
5	21
6	112
7	853
8	11,117
9	2,61,080
10	1,17,16,571

The patterns are repeated subgraphs found in a network. But discovering the patterns in a particular network is really a challenging task. The task get tougher with the large datasets and label graph. The problem is the sheer number of distinct topologies get generated even considering the smaller number of vertices (see Table 1). As the number of patterns increases exponentially with respect to the size of vertices it makes mining task tougher. While discovering patterns in a network one has to take care of different isomorphic form the pattern get generated even for a small number of vertices. To check isomorphism, the canonical form helps to differentiate the topologies. But even same pattern has many canonical forms. Because of this discovery pattern in a single graph computationally expensive. There is no polynomial equation available to counter this problem which makes it as an *NP*-Complete problem.

To counter the computationally expensive task, the approximation method is better to solve real-life practical problem [7]. As some of the approximation methods like Markov Chain Monte Carlo (MCMC) have a low error rate. Which gives perfect reason to use it. So the sampling based technique with minimal error rate is the best way to solve the problem and use it in to study real-life biological network. Biological network motif has the range of applications like with the help of machine learning classifying network into different superfamilies [8], discovering essential protein in PPI networks [9].

The rest of the paper is organized as follows. In section 2 we discuss some relevant research work of this field. Section 3 describe the problem definition which are used to understand the proposed work. Section 4 has a proposed method in which we took one example to explain how algorithm works. Experiment and results are on the section 5 and comparison results with previously proposed method. Last section 6 has a conclusion and future scope.

2. Related Work

Recently graph is used for modeling the complex structures like protein structure, biological network, chemical compound and social network. Graphs are the most powerful data structure for the representation of object, concepts and their relationship. Graph mining has gained substantial importance in data mining field. Many researchers propose novel methods and algorithm for graph mining to find the frequent pattern in a particular network.

Network with the similar global property may have the different local structure. The local properties of any network are the building block of any network and it shows the actual fingerprint of the network. The local structure may have a particular pattern which has substantial importance to building the network. Also, the distribution of pattern with respect to local structure and global structure helps for many applications in the biological network.

Milo et al. [6] is the pioneer for the network motifs which are frequently found in biological networks. The motif defines as the particular pattern found in an inputted graph with a significantly higher number of frequency found as compared to randomly generated graph with a sample number of vertices as an inputted graph. The pattern with high significance play an important role because interaction follows certain properties and pattern capture it. It helps to apply a large number of a real-life application involving the study of the biological network. The researcher started proposing different approaches and methods for finding motifs in a different biological network and improving performance and accuracy.

Kim et al. [10] use network motif to find the frequent patterns in the large biological network. As the size increases the complexity of problem increases. The new novel based algorithm has been proposed to improve the performance and get high accuracy rate. The motif further uses to study different biological problems like cellular biology, gene structure, and cell behavior. Firana et al. [11] use network motif to understand the molecular mechanism in the cell behavior and proposed the new algorithm to detect motif efficiently with the high speed.

In last decade many researcher use network motifs to study the large graph and their relationship and infer many important results which may use in the decision making in a classification. Various methods and tools have been proposed to detect the motif in the network and give the solution to the computationally challenging problem like MFinder [12], FANMOD [13], Kavosh [14] and NETMODE [15]. But all these methods are not computationally powerful with respect to a size of network and pattern looking for. As the size of pattern goes increases the performance gets decreases. Also, these methods do not scale very well with the size of the network. The large graph network takes hours to process with low accuracy rate.

In counting frequent patterns and their frequencies, subgraph isomorphism task makes more challenging because of computation power it requires to check with canonical labels. To overcome this problem many researchers use approximation approaches. As the approximation does not guarantees the exact match but it helps to reduced computation cost. Some of the best approximation technique is sampling. As the sampling guarantees to minimum error rate [16]. It makes more sense to use it. The performance of sampling based method depends on the accuracy and running time.

3. Problem Definition

The graph represented as $G(V, E)$, where V is a collection of vertices and E is a collection of edges. As edge shows the interaction between vertices so, edge $e \in E$ represents pair having two vertices (v_i, v_j). We are considering only undirected graphs so graph supposes to be simple, connected with no self-loop and no multi-edges.

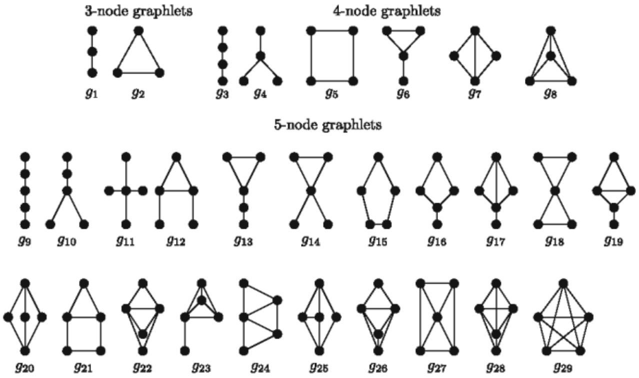

Figure 1: All 3, 4, 5 vertex distinct graph patterns

The patterns are the small embedded subgraphs in a network. A graph $S = (V', E')$, S is a subgraph of G where, $V' \in V$ and $E' \in E$. The pattern considered to be induced subgraph. The induced subgraph is a subset of vertices with edges whose endpoint are both in this subsets. If pattern S is an induced subgraph of graph G and also $|V'| = p$, then we can call the subgraph is a p-pattern of G. Also our pattern follows distinct topologies show in Figure 1. Patterns are based on the vertices number which are fixed number of distinct topologies. We maintain set of patterns denoted as Λ_p where p is a distinct topology. As the pattern depends on the number of vertices and for different vertices has many different pattern so specific pattern in a set Λ_p, we call as $\omega_{p,q}$ where q is an arbitrary order of particular topology. So the Λ_p is a set of embedded subgraphs of graph G. Figure 1, shows the pattern set as Λ_3, Λ_4 and Λ_5. The order in which they appear, we can extract as $\omega_{3,2}$ is a triangle form of a pattern.

3.1. Concentration of pattern

The concentration depends on the frequencies of p-pattern it appears in a graph G. The distinct pattern denoted as s and their frequency as $F_G(s)$. The concentration of particular p-pattern $C_G(s)$ defined as a ratio of pattern appeared in a graph to the total pattern in a set Λ_p, which formulated by,

$$C_G(s) = \frac{F_G(s)}{\sum_{i \in \Lambda_p} F_G(i)} \tag{1}$$

3.2. Motif

Motifs are the patterns found in a biological network which are more frequent [6]. The patterns mainly compared with the random graph model also called as the null model. The pattern has higher appearance frequency than compared to the randomly generated network. As the randomly generated graph follow same degree distribution, which prove particular pattern has great statistical significance.

3.3. Morkov Chain

A morkov chain is a sequence of processes that experience transition of states from sample space S completely depends on the probability they appear. The transition of states fully depends on the transition probability matrix. The probability distribution is denoted as $\pi(i)$ and proposed distribution as q. The markov chain is the memoryless process because future states do not depend on the state that led up to the current state. The chain is called to be stationary if transition process always has a future state despite how current state arrived. That means markov chain is *ergodic*. Also, chain satisfies reversibly condition $\pi(x)T(x,y) = \pi(y)T(y,x), \forall_{x,y} \in S$. The main aim of the algorithm is to draw sample x from sample space with target distribution. Here we are using the Metropolis-Hasting (MH) algorithm for acceptance of samples. The target distribution is $\pi(x) = f(x)/K$, where, K is some constant value needs to set which required domain expertise. Acceptance probability depends on the equation:

$$\propto (x,y) = \min\left(1, \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right) \tag{2}$$

4. Proposed method

We are taking input graph G as an undirected graph and p size pattern. The sampling method sample the different set of p-pattern of input graph G. By enumerating all the patterns of given size the concentration for the particular pattern can be calculated with the help of Equation 1. As a sampling is a process of random walk. The sample-based method must perform in an unbiased way. Otherwise, the higher degree vertices get favor and expected probability distribution will not be the same. The enumeration task gets tougher with larger sample space which will get generated. So to counter the real-life problem direct method cannot be used, it required enumeration of all millions of pattern. This task will take lots of time which, exactly what we need to avoid. So it makes perfect sense to use indirect sampling based method. Many previously proposed method uses indirect sampling like MFinder [12] and FANMOD [13] to boost their performance. But the problem with these previous methods is like MFinder need to require adjusting the concentration to correct bias. On the other hand FANMOD method, no needs to adjust concentration and also guarantees the uniform sampling but the task is costly.

We are proposing Frequent Pattern Search (FPS) algorithm to sample p-patterns from input graph G using Markov Chain Monte Carlo (MCMC) sampling. As the markov chain required random walk over the state's space. The chain guarantees the stationary distribution which leads to converge the desired target distribution. Metropolis-Hasting helps with the acceptance of states. Here states represented a set of p-pattern. To accept target distribution we considered to be uniform [16] i.e. we want to sample each p-pattern with the identical probability. If P is collecting all p-pattern and p is our target distribution then $\pi(s) = 1/|P|, \forall s \in P$.

FPS algorithm performs random walk all over the input graph. To traverse the neighboring pattern one has to select randomly a vertex from neighboring vertices pool and replace it with new vertices but it needs to be vertices induced subgraph. That means, for every iteration new neighbor vertices get to discover and that's how we traversing the whole graph. Also, discuss implementation issues.

4.1. Neighboring Pattern

Consider one example for our method Frequent Pattern Search (FPS) to demonstrate how neighboring pattern gets selected.

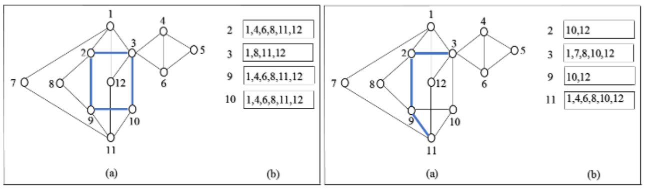

Figure 2: Neighbor generation

From Figure 2(a) (Left) we can see a bold line as a 4-pattern $\langle 2,3,9,10 \rangle$ which is existing state reached by the random walk. To perform next iteration of random walk by selecting of the neighboring vertices from vertices pool on Figure 2(b) (Left). Newly neighbor pattern gets generated by deleting old vertex by the newly added vertex. In Figure 2(a) (Right) we can see vertex 11 gets picked randomly by deleting vertex 10 so the neighbor pattern is $\langle 2,3,9,11 \rangle$. Updated neighbor updates its information on Figure 2(b) (Right) block which is neighboring vertex pool. As the degree of each state is very important for random walk probability calculation which is a neighbor count of pattern, which is the sum of all entries in boxes. So the degree of pattern $\langle 2,3,9,10 \rangle$ is 22 and neighbor pattern $\langle 2,3,9,11 \rangle$ is 15. The degree we need to calculate the probability.

4.2. Frequent Pattern Search Algorithm

Algorithm 1 Frequent Pattern Search (FPS) Algorithm

```
        $G$: Graph
        $p$ =Pattern Size
        $N$ = Iteration Number for Sampling $|S|$
$s$= Initial State
$M$=$\emptyset$
$i$=0
$d_s$=Number of neighbors of $s$
while $i < N$ do
        $x$=Randomly selected neighbor of $s$ from (1,$|d_s|$)
        $d_x$=Number of neighbors of $x$
        $probability$=$d_s/d_x$
        $accept\_probability$=min(1,$probability$)
        if $uniform\_dist(0,1) \le accept\_probability$ then
                $s = x$
                $d_s = d_x$
        $i = i + 1$
        Make Canonical forms of $s$
        Put Canonical forms into the set $M$
        Update pattern count
Normalize the frequency
$\forall_j w_{p,j} \in M$
return $M$
```

The Frequent Pattern Search (FPS) Algorithm mention in Algorithm1. Here FPS use metropolis-hasting algorithm to decide when state to be accepted for which we need to decide proposal distribution q. For our algorithm, we decided to propose distribution to be uniform [16]. If $n \in P$ having n as a neighbor of s as per proposed neighbor state. Using propose distribution, the selection of n from s is decided by probability value which is calculated by $q(s,n) = 1/d_s$, where d_s is a degree of state of s. Also if $m \in P$ where m is not a neighbor of s, $q(s,m) = 0$, we decline the state.

We have implemented our algorithm by using proposal (q) and target (p) distribution. The sample size is decided with the iteration number accept from the user. Respective pattern count is stored in memory for each pattern in Λ_p. For each iteration, we identify a pattern and updates its counter by 1. We normalized the frequency by considering concentration mention in Equation 1. Thus, if S is a sample set and $s \in \Lambda_p$ mathematically:

$$\hat{C}(s) = \frac{1}{|S|} \sum_{i \in S} 1_{(i==s)} \tag{3}$$

The method chooses one of its neighbor in each iteration, using the proposed distribution which is uniform we decide the acceptance or rejection of state. This is how our algorithm adjusts the transition probability and guarantees the target distribution to be uniform.

4.3. Implementation

4.3.1. Initial State

The algorithm uses the random walk to explore the vertices on G. Algorithm starts with arbitrary p-pattern which we called vertex induced subgraph. To find neighbor pattern vertex gets pick by the random process from vertices set. Newly selected vertex get populate other neighboring vertices. And process gets repeat for the number of iterations. As the process always return p-pattern with the same number of vertices as p.

4.3.2. Isomorphism Check

One of the time-consuming tasks is to check isomorphism. It required high computation power because we need to generate different canonical form. Every pattern has generated multiple canonical forms with the same number of vertices. But the minimal canonical form of two patterns always be the same. So for each pattern always needs to check canonical form and also figure out its minimal lexicographical order. Which adds up a load on computation and make a time-consuming task. We are using *min-dfs-code* [17] to check isomorphism in our algorithm.

4.3.3. Queue Management

The number of patterns changes with respect to vertices of size (see Table 1). So at every iteration, we need to check different pattern from the queue. Lookups for pattern adds large computation. We implemented a special queue with the help of multi-index map data structure. This special boosting library provides many functionalities like indexing, sorting with the high speed. To order canonical label in lexicographically this data structure is really helpful. It reduced the computation power required to check isomorphism and improve performance in execution time. This process makes our proposed work to be better than existing methods.

5. Experiments and Results

We have implemented Frequent Pattern Search (FPS) algorithm in C++ language with the help of different libraries like multi-index container, random, graphs, string tokenization etc. We execute the experiment on the computer with 3.4 GHz processor with 4 GB RAM. We choose Linux operating system to run our code. Also, perform different experiments on the Protein-Protein Interaction (PPI) datasets. The experiment has done with different size of datasets collected from Database of Interacting Protein (DIP) [18]. Table 2 shows the list of various datasets with a different number of vertices and edges. Also, an average degree helps to understand network better. Since we are only performing experiments on the undirected and simple graph so we made datasets to be undirected, connected and with no self-loop.

Table 2: Graph Data Statistics

Graph		Vertex	Edge	Average Degree
S. cerevisiae	Scere20170205	5,176	22,977	8.878
	Scere20160114	5,166	22,914	8.871
E. coli	Ecoli20170205	2,940	12,261	8.340
	Ecoli20160114	2,938	12,252	8.340
High-Throughput DIP	Li2004a	2,633	4,027	3.058
	Giot2003a	7,036	20,926	5.939
	Gavin2002a	1,361	3,222	4.734

Many methods have been proposed to do this task. Wernicke shows in his research that the FANMOD [13] is better than MFinder [12]. The FANMOD is faster with better accuracy rate. Recently FANMOD also shows the better result than MODA [19]. With respect to a various performance metric, we can say that FANMOD is one of the best methods as compared to other. So it makes perfect sense to compare our methods with FANMOD. We use available implementation of FANMOD Library to compare our results with. It has a limitation on a size of the pattern which is up to 8 vertices only. Also while using sampling method we need to set some probability values. We refer FANMOD documentation and set recommended values of probability.

5.1. Error Rate

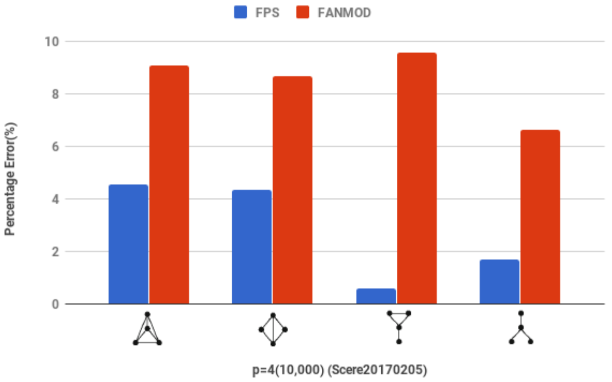

Figure 3: Comparison of Percentage Error p=4

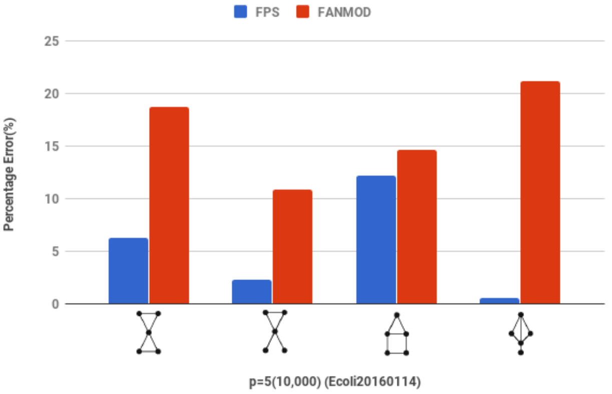

Figure 4: Comparison of Percentage Error p=5

To compute the error rate we first need to find an exact concentration of particular pattern using an exact method in G. Second, now compute concentration by using our approximation methods. The error is the ratio of an absolute difference between above two concentration to the exact method concentration for particular pattern p. Our method use sampling which requires random access. While calculating concentration we take an average of 5 different runs.

We compare Percentage Error with FPS and FANMOD for different datasets with different pattern size. We compare results with few patterns which are likely to be the motif. To compare result, sample size to be fixed with 10,000 iteration and two DIP datasets with the different number of vertices and edges. Figure 3, shows the error comparison on Scere20170205 dataset having pattern size to be 4. And Figure 4, shows the error comparison on Ecoli20160114 datasets having pattern size to be 5.

By analysing result we can say that proposed method is way better than FANMOD. With very less error rate. This is not always true statements sometimes our methods also give a bad result as FANMOD but the majority results are really promising.

5.2. Execution time

Execution time comparison is shown in Table 3 with FPS and FANMOD. Here a number of samples are fixed 10,000 for all the datasets to both the methods. As we can see execution time which is increased with respect to the pattern size in both the methods. But the FANMOD shows the poor scalability with increased number of pattern and graph size. Clearly, FPS has better execution time.

Table 3: Execution Time Comparison

Dataset	**Pattern Size**	**FPS** (Sec)	**FANMOD** (Sec)
Scere20170205	3	1.12	0.26
	4	1.43	6.42
	5	2.73	441.34
Scere20160114	4	1.44	7.55
	5	2.60	476.53
Ecoli20170205	3	0.66	0.86
	4	1.38	3.97
	5	2.44	159.00
Ecoli20160114	4	1.40	4.07
	5	2.50	137.07
Li2004a	4	1.10	0.952
	5	2.18	12.84
	6	4.62	433.60
Giot2003a	4	0.88	1.87
	5	1.65	2.66
	6	3.29	1648.83
Gavin2002a	6	1.55	7.254
	7	2.56	184.64
	8	6.8	2005.90

Figure 5, shows the execution time performance metric with respect to increase iteration of the sample. To perform experiment we take Ecoli20160114 dataset and calculate execution time for 4 size pattern. The graph shows that if the number of iteration increases, execution time also get increases in both methods. Overall time FPS take to execute is very less.

Figure 6, shows the execution time performance metric with respect to pattern size. As FANMOD method only supported up to 8 vertices size pattern. We considered pattern size from 3 to 10. The execution performs on smallest dataset Gavin2002a from our collection. It is clear from the plot that FPS scale very well with an increased number of patterns. For the FANMOD execution time grows with increase pattern size.

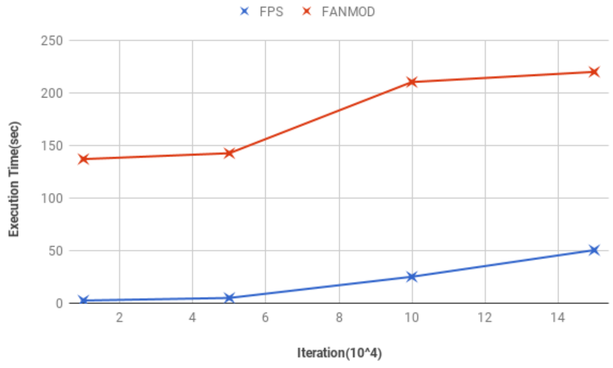

Figure 5: Iteration v/s Execution Time

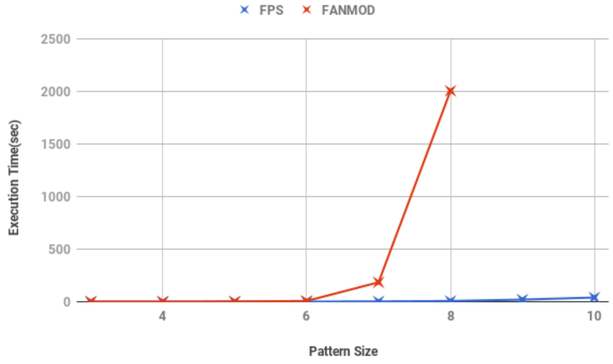

Figure 6: Pattern Size v/s Execution Time

6. Conclusion

In this paper, we proposed an algorithm called Frequent Pattern Search (FPS), which is based on sampling method. We find different $\boldsymbol{p}$-pattern and their frequency in a Protein-Protein Interaction (PPI) network datasets. Our experimental results demonstrates that our proposed method is significantly faster than the best of the existing method. Also, result show the lower error rate, that means proposed method is reliable. This method can be used to perform different experiment with large size biological datasets and different domain datasets also.

References

[1] Albert, István, and Réka Albert. "Conserved network motifs allow protein–protein interaction prediction." *Bioinformatics* 20.18 (2004): 3346-3352.

[2] Callebaut, Werner. "Scientific perspectivism: a philosopher of science's response to the challenge of big data biology." *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences* 43.1 (2012): 69-80.

[3] Chen, Jin, et al. "Labeling network motifs in protein interactomes for protein function prediction." *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007.

[4] Kim, Wooyoung, and Lynnette Haukap. "NemoProfile as an efficient approach to network motif analysis with instance collection." *BMC bioinformatics* 18.12 (2017): 423.

[5] Milenković, Tijana, and Nataša Pržulj. "Uncovering biological network function via graphlet degree signatures." *Cancer informatics* 6 (2008): CIN-S680.

[6] Milo, Ron, et al. "Network motifs: simple building blocks of complex networks." *Science* 298.5594 (2002): 824-827.

[7] Saha, Tanay Kumar, and Mohammad Al Hasan. "Finding network motifs using MCMC sampling." *Complex Networks VI*. Springer, Cham, 2015. 13-24.

[8] Milo, Ron, et al. "Superfamilies of evolved and designed networks." *Science* 303.5663 (2004): 1538-1542.

[9] Kim, Wooyoung, et al. "Essential protein discovery based on network motif and gene ontology." *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*. IEEE, 2011.

[10] Kim, Wooyoung, et al. "Biological network motif detection and evaluation." *BMC systems biology* 5.3 (2011): S5.

[11] Farina, Lorenzo, et al. "Identification of regulatory network motifs from gene expression data." *Journal of Mathematical Modelling and Algorithms* 9.3 (2010): 233-245.

[12] Kashtan, Nadav, et al. "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs." *Bioinformatics* 20.11 (2004): 1746-1758.

[13] Wernicke, Sebastian, and Florian Rasche. "FANMOD: a tool for fast network motif detection." *Bioinformatics* 22.9 (2006): 1152-1153.

[14] Kashani, Zahra Razaghi Moghadam, et al. "Kavosh: a new algorithm for finding network motifs." *BMC bioinformatics* 10.1 (2009): 318.

[15] Li, Xin, et al. "Netmode: Network motif detection without nauty." *PloS one* 7.12 (2012): e50093.

[16] Bhuiyan, Mansurul A., Mahmudur Rahman, and M. Al Hasan. "Guise: Uniform sampling of graphlets for large graph analysis." *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012.

[17] Yan, Xifeng, and Jiawei Han. "gspan: Graph-based substructure pattern mining." *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002.

[18] Xenarios, Ioannis, et al. "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions." *Nucleic acids research* 30.1 (2002): 303-305.

[19] Omidi, Saeed, Falk Schreiber, and Ali Masoudi-Nejad. "MODA: an efficient algorithm for network motif discovery in biological networks." *Genes & genetic systems* 84.5 (2009): 385-395.