34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

A Network Clustering Algorithm for Detection of Protein Families

Jiang Xie, Minchao Wang, Dongbo Dai, Huiran Zhang and Wu Zhang

Abstract—**Detection of protein families in large scale database is a difficult but important biological problem. Computational clustering methods can effectively address the problem. Although there exist many clustering algorithms, most of them are just based on the threshold. Their computational performances are affected by the weight distribution greatly, and they are only valid for some special networks. A new network clustering algorithm, Markov Finding and Clustering (MFC), is proposed to cluster the proteins into their functionally specific families accurately in this paper. The MFC algorithm makes an improvement in the random walk process and reduces the affection of the noise on the clustering result. It has a good performance on these networks which are not well addressed by existing algorithms sensitive to the noise. Finally, experiments on the protein sequence datasets demonstrate that the algorithm is effective in the detection of protein families and has a better performance than the current algorithms.**

I. Introduction

In the past decade, many advanced biological technologies have been introduced into the biology research. Information technology is one of the advanced technologies, and has been widely used to address the biology problems because of its powerful calculation capacity. Moreover, the Genome projects [1] bring the explosion in the available sequence data. Currently, the UniProt database [2] contains about 20 million sequence entries and compared with the sequence amount in 2011, it makes about 2 times increment. However, a large proportion of these protein entries have not been experimentally characterized, so it is difficult for us to determine their functions or biological processes. It is well known that proteins with the same functions or biological processes should be in the same protein families [3] and the protein families are defined as the groups of molecules which share significant sequence similarity [4]. Hence, it is possible for us to detect the protein families through the protein sequences similarity relationships, and these relationships can be obtained easily nowadays by some existing approaches such as BLAST [5]. In order to detect the protein families, we should take into account all similarity relationships and it is very hard for us to handle these large scale data manually. Computational clustering methods can address these challenges in an efficient way. Most of recent algorithms about the network clustering aim to cluster a similarity network which is generated from the protein sequences similarity relationship. In the similarity network, vertices present the proteins and edges present the sequence similarity relationship between the proteins.

Network clustering algorithms are much different from the traditional clustering algorithms such as K-means. Network clustering algorithms don't need the prior knowledge and cluster each protein into only one protein family. In recent years, there have been many network clustering algorithms emerging such as TRIBE-MCL [7], SCPS [8] and FEC [9]. In general, most network clustering algorithms can be grouped into two classes: geometry-based and flow-based [6]. MCL and FEC are the flow-based algorithms, and SCPS is the geometry-based algorithm.

The TRIBE-MCL algorithm proposed by Enright,A.J. is based on the Markov chain and can cluster the network into different segments without the prior knowledge [7]. TRIBE-MCL algorithm takes the sequence similarity matrix and a coefficient r as input and iteratively runs two operations, expansion and inflation. Expansion operation takes the power of a matrix and each expansion operation simulates random walks with many steps in the network. Inflation operation is to take the Hadamard power of a matrix, and the parameter r is the power coefficient which controls the 'tightness' of the clusters. With the iteration running, the relationships of the intra-cluster are promoted and the relationships of the inter-cluster are demoted. After a few iteration steps, the network will be separated into different segments and each segment presents one cluster.

SCPS algorithm proposed by Alberto et al. is a geometry-based algorithm to address the network clustering problem by the spectral methods [8]. The algorithm is also based on the random walk, but it processes the normalized graph Laplacians matrix instead of the probability matrix. It clusters the network by analyzing the eigenvectors of the similarity matrix, so it is always based on the stationary distribution of the matrix during clustering. This advanced characteristic makes the clustering result more accurate than the TRIBE-MCL algorithm, but it needs more running time.

Network clustering algorithm is also be widely used in the social networks. FEC algorithm presented by Yang, B. et al. is to find the community in the signed social network [9]. It is to extract the community structure by sorting the probability of each node reaching a destination node, named "sink node", after a few random walks. The nodes with the higher probability are more likely to be in the same clusters with the sink node, and the community can be extracted from the sorted

Research supported by Key Project of Science and Technology Commission of Shanghai Municipality [No.11510500300], Shanghai Leading Academic Discipline Project [No.J50103], Innovation Program of Shanghai Municipal Education Commission [No.11YZ03] and Ph.D. Programs Fund of Ministry of Education of China [No. 20113108120022].

Jiang Xie is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China and the Department of Mathematics, University of California Irvine, Irvine, CA 92697, USA.

Minchao Wang is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China.

Dongbo Dai is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China.

Huiran Zhang is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China.

Wu Zhang is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China and the High Performance Computing Center, Shanghai University, Shanghai 200072, China(corresponding author to provide e-mail: wzhang@shu.edu.cn)

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

probability distribution by the signed cut method. The core idea in the FEC algorithm is that nodes can be more easily to be reached by those nodes which are in the same cluster.

In order to improve the performance of the existing network clustering algorithms, Leonard et al. proposed an approach which detects the edge weight distribution of network and sets the different thresholds corresponding to different network [10]. Leonard also applied thresholds into the MCL and SCPS algorithms and found that the performances of both algorithms have a great improvement with the thresholds.

However, for the MCL algorithm, the random walk is based on the current iterating result, so its performance is much affected by the strong inter-cluster relationship at the beginning of the random walk. Moreover, both MCL and FEC algorithms are very sensitive to the noise, so they don't have a good performance on the networks with some noises. Though the SCPS algorithm has a great performance, it needs a long running time.

Compared with these algorithms mentioned above, the Markov Finding and Clustering (MFC) algorithm proposed in this paper, integrates the core ideas of the MCL and FEC algorithms. The algorithm makes an improvement in the performance. Our contributions are as follows:

- Making a modification on the random walk process and reducing the affection of the strong inter-cluster relationship at the beginning of the random walk.
- Taking both probabilities of the 'to' and 'from' into account (the 'to' probability means the probability of the source node to the target node, the 'from' probability means the probability of the target node to source node) and making the clustering performance better and less sensitive to the noise.

The rest of this paper is organized as follows. In section II, the detailed description of MFC algorithm is presented. In section III, we validate the MFC algorithm through the experiments on the protein sequence datasets. Finally, we discuss and summarize our works in section IV.

II. Method

A. *Similarity network*

We construct similarity networks by carrying out all-against-all BLAST search using local databases built from each protein sequence datasets and detect the protein families through clustering the similarity network. In the similarity network, nodes represent the proteins and edges represent the sequence similarity relationship between nodes. We set a weight, which is equivalent to the –log of the BLAST *E-value* on each edge. In our experiments, the cutoff of the BLAST is all set to 1e-10.

B. *Algorithm*

For a network with the cluster structure, edge density of the intro-cluster is much higher than that of the inter-cluster, so it is more likely for a 'walker' to reach the other nodes in the same cluster. Moreover, according to the FEC algorithm [9], we know that a "sink" node has higher probability to be reached from the other nodes which are in the same cluster than these are not. Based on the both ideas mentioned above, we propose the MFC algorithm which includes three operations: walking, finding and inflation.

1) *Walking operation*

Walking operation is to simulate the random walk in the probability network. The probability network is generated from the similarity network and all data of the probability network are stored in a probability matrix. Each row or column represents one node in the network and each element in matrix represents the moving probability between nodes. For example, the element A_{ij}, which is on the row *i* and column *j*, represents the probability of moving from node *i* to node *j*. In fact, the probability matrix is equivalent to the probability distribution of one step random walk.

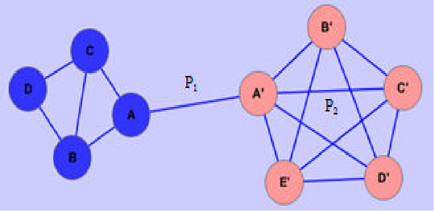

Figure 1. Two clusters identified by the blue nodes and the red nodes. P_1 represents the probability of A' to A and P_2 represents that of A' to C, respectively. The P_1 and P_2 probabilities are generated from the E-value of the BLAST.

Compared with the expansion operation in the MCL, the walking operation in the MFC is always based on the stationary distribution of the initial probability matrix. At the beginning of random walk, the walk length is short and the walking probabilities between the intro-cluster nodes are not all higher than these between the inter-cluster nodes. After the inflation (described in *3)*), the strong inter-cluster edges will be promoted. If the random walk is based on the current iterating result, these inter-cluster edges which have been promoted will become stronger in the next step random walk and have a great influence on the later steps.

For example, there are two clusters in the Fig. 1. At the first step of the random walk, the walk length is one and only one edge can be chosen by A' to move to A or C'. The probability of A' moving to A and C' is P_1 and P_2, respectively. And P_1 is higher than P_2. After the inflation, P_1 is promoted and P_2 is demoted. For MCL algorithm, because the second expansion is based on the result of the first random walk, the performance of the second step random walk will be affected and this affection will exist in the later expansion until the iteration is finished. However, for MFC algorithm, because the walking operation is based on the initial probability matrix, the affection of P_1 will be eliminated gradually. Because with the walk step length becoming longer, A' has much more intro-cluster edges than inter-cluster edges to choose. For example, when the walk step length is 2, A' can move to C' through three paths, A' B' C', A'D'C' and A'E'C', while the path between A' and A is still one. So the probability of A' to C' becomes higher than probability of A' to A gradually.

2) *Finding operation*

Finding operation is to find clusters in the network. It includes two steps. The first step is to make a summation. Because the node pairs in the same cluster have higher probability to reach each other, we can detect these kinds of node pairs by making the summation of the two probabilities. For example, nodes i and j are one pair nodes in the same cluster, so both of the probabilities A_{ij} and A_{ji} are high and the summation of them will be high as well. We can get the summation results of all node pairs by summating the probability matrix and its transposed matrix. The equation is as (1) and the *SA* in the equation is the result matrix after summation.

$$SA = A + A^T \tag{1}$$

The second step of finding operation is to revise the deviation. Sometimes there are some noises on weight of edges and these noises cause a great deviation between 'to' and 'from' probabilities, which makes a great affection on the performance. In order to eliminate these affections, we revise the deviation between the two probabilities. So in the second step, we take subtraction of A^T from A. The equation is as (2) and *RA* is the result matrix after subtraction.

$$RA = | A - A^T | \tag{2}$$

After the two steps, the result matrix of finding operation can be obtained by (3).

$$FA = SA - RA \tag{3}$$

Compared with some other algorithms, we take both two probabilities, the 'to' and 'from' probabilities, into account in MFC algorithm. If only one probability is considered, some noises between clusters will influence the performance.

Given a probability matrix A, there is a noise on the edge from node i to node j, so the probability A_{ij} is much higher than A_{ji}. So it is more easily to walk from node i to node j. These algorithms which only consider the 'to' probability will cluster node i and node j together. However, sometimes they are not in the same cluster. However, for the MFC algorithm considering both 'to' and 'from' probabilities, the noise in the A_{ij} does not influence the performance of algorithm. Though it is easily to walk from node i to node j, it is hard for the node j to leave its own cluster, because the intro-cluster edges are much denser.

3) *Inflation operation*

Inflation operation is to extract the clusters in the network. After the inflation operation, the weak relationships between the clusters are demoted and the strong relationships in the clusters are promoted. The inflation operation in the MFC algorithm is the same as it in the MCL algorithm. It takes the Hadamard power of the probability matrix and the coefficient r controls the 'tightness' of the clusters. Given a probability matrix A with the order of n, the equation of inflation operation is as (4).

$$(\Gamma_r A)_{pq} = (A_{pq})^r / \sum_{i=1}^{n} (A_{pi})^r \tag{4}$$

A_{pq} is the element on the row p and column q of probability matrix and represents the probability of moving from node p to node q. The $(\Gamma_r A)_{pq}$ corresponds to the value of A_{pq} after the inflation operation.

With the iteration running, the weight of inter-clusters edges become weaker and the weight of intro-clusters edges become stronger gradually. When the probability matrix does not change any more with the further iteration, the iteration stops and the weak edges which are out of the computer precision are eliminated. So the cluster structure in the network appears automatically.

III. Experiments

In this section, in order to validate our algorithm, we test the algorithm on three protein superfamily datasets.

A. *Protein sequence datasets*

Proteins in the same family are not only more similar with each other in the sequences, but also they are more likely to have the same function in the biological process. In this paper, we select three superfamilies, namely Enolase[11], Crotonase[12] and Amidohydrolase [13] from the Structure-Function Linkage database [14] to validate our algorithm. These superfamilies are regarded as the gold-standard to test our clustering performance and there are total 904 Enolase sequences, 452 Crotonase sequences and 2075 Amidohydrolase sequences are in each protein superfamily, respectively.

B. *Protein superfamily clustering*

Through the experiments, we demonstrate the validation of the MFC algorithm. Networks are separated into different clusters and are visualized by the Cytoscape's [15] Organic layout, force-directed layout algorithm.

TABLE I. The accuracy of each cluster of the MFC algorithm on the Enolase superfamily dataset The first row (**Total**) and the second row (**Wrong**) means the total protein number and the wrong protein number of each cluster, respectively.

Total	45	297	35	19	246	7	65	190
Wrong	25	0	7	1	0	4	0	0
Accuracy (%)	44.4	100	80	94.7	100	57.1	100	100

Table I shows the detailed clustering result on the Enolase superfamily. The superfamily consists of 8 protein families. From the analysis on this result, we find that 4 big proteins families are clustered exactly; some small protein families are separated into different smaller clusters and most of these smaller clusters always just contain 1 or 2 nodes. Moreover, we also calculate the F-measure of the result and its value is about 0.9577.

The clustering result of the Amidohydrolase superfamily is presented in the Fig. 2 and the detailed result is presented in the Table II.

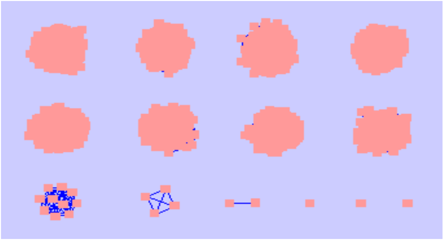

Figure 2. Visualization of the clustering result of the MFC algorithm on Amidohydrolase superfamily. Each square node represents a protein and each set represents a cluster. The superfamily is separated into 14 clusters.

TABLE II. THE ACCURACY OF EACH CLUSTER OF THE MFC ALGORITHM ON THE AMIDOHYDROLASE SUPERFAMILY DATASET. THE FIRST ROW (**TOTAL**) AND THE SECOND ROW (**WRONG**) MEANS THE TOTAL PROTEIN NUMBER AND THE WRONG PROTEIN NUMBER OF EACH CLUSTER, RESPECTIVELY.

Total	249	260	261	415	113	299	300	178
Wrong	2	4	1	1	1	10	0	0
Accuracy (%)	99.2	98.5	99.6	99.8	99.1	96.7	100	100

From the table, we find that only 19 nodes are separated into the wrong clusters and the accuracy of each cluster is about 99%. In addition, we calculate the F-measure value to demonstrate the validation of algorithm. The best F-measure value of MFC on this dataset is 0.995.

TABLE III. THE ACCURACY OF EACH CLUSTER OF THE MCL ALGORITHM ON THE AMIDOHYDROLASE SUPERFAMILY DATASET. THE FIRST ROW (**TOTAL**) AND THE SECOND ROW (**WRONG**) MEANS THE TOTAL PROTEIN NUMBER AND THE WRONG PROTEIN NUMBER OF EACH CLUSTER, RESPECTIVELY.

Total	249	260	261	415	113	299	300	178
Wrong	0	61	0	1	0	113	0	0
Accuracy (%)	100	76.25	100	99.8	100	62.21	100	100

In order to confirm the improvement in the performance, we test the MCL algorithm on this same protein superfamily by using the MCL tool which is developed by Stijn van Dongen [16]. In table III, we list the detailed accuracy of each cluster and find that 175 nodes are separated into the wrong clusters. The best F-measure value and the total accuracy of this result are just about 0.94 and 91.6%, respectively.

The third dataset used to validate our algorithm is the superfamily Crotonase, which contains 452 protein sequences and 6 protein families. Comparing the performance of both algorithms on this dataset, we find that the MFC algorithm is also better than the MCL algorithm. The best F-measure value is 0.9778, which is obtained when the superfamily is separated into 7 families.

IV. CONCLUSION

In this paper, we propose an algorithm which is based on the Markov chain and simulates the random walk in the network. Compared with MCL or FEC algorithm, MFC takes the both ‘to’ and ‘from’ probabilities into account, so it can reduce the influence of the noises effectively and perform better in clustering. We analyze the experiment results and find that the Finding Operation plays a key role in the improvement. In each iterative step, the Finding Operation makes the probabilities of random walk more credible and reduces much influence of the ‘false relationship’ between nodes caused by the noise. In addition, we revise the random walk in the MFC algorithm in which we make the random walk based on the initial probability distribution instead of the current iterating result. After the revise, each walking step in the network becomes much steadier and less sensitive to the ‘false relationship’ or the noise at the beginning of the random walk process. Finally, we test our algorithm through experiments and demonstrate that our algorithm has a good performance in the detection of protein families.

ACKNOWLEDGMENT

The authors wish to thank for the discussion with the bioinformatics group of the School of Life Science, Shanghai University, and the first author is grateful to the visiting University of California at Irvine and working on Computational Biology with Professor Qing Nie, Centre for Mathematical and Computational Biology, University of California at Irvine.

REFERENCES

[1] Bernal,A., Ear,U. and Kyrpides,N. Genomes online database(GOLD): a monitor of genome projects world-wide[J]. Nucleic Acids Res, 2001, vol.29(1):pp.126-127.

[2] Apweiler,R. and Bairoch,A. UniProt: the universal protein knowledge base[J]. Nucleic Acids Res, 2004, vol.32(1):pp.115-119.

[3] Hegyi,H. and Gerstein,M. The relationship between protein structure and function: a comprehensive survey with application to the yeast genome[J]. J.Mol.Biol, 1999, vol.288(1):pp.147-164.

[4] Dayhoff,M.O. The origin and evolution of protein superfamilies[J]. Fed. Proc, 1976, vol.35(10):pp.2132-2138.

[5] Altschul,S.F., Gish,W. and Miller,W. Basic local alignment search tool[J]. J. Mol. Biol, 1990, vol.215(3):pp.403-410.

[6] Frivolt,G. and Pok,O. Comparison of graph clustering approaches[C]. In Proceedings in IIT.SRC. Slovak University of Technology, Veliko Turnovo, Bulgaria, pp.168-175.

[7] Enright,A.J., Van Dongen,S. and Ouzounis,C.A. A efficient algorithm for large-scale detection of protein families[J]. Nucleic Acids Res, 2002, vol.30(7):pp.1575-1584.

[8] Paccanaro,A., Casbon,J.A. and Saqi,M.A.S. Spectral clustering of protein sequences[J]. Nucleic Acids Res, 2006, vol.34(5):pp.1571-1580.

[9] Yang,B., Cheung,W.K. and Liu,J. Community mining from signed social networks[J]. IEEE Trans. Knowl. Data En, 2007, vol.19(10):pp.1333-1348.

[10] Apeltsin,L., Morris, J.H and Babbitt.P.C. Improving the quailty of the protein similarity network clustering algorithms using the network edge weight distrubution[J]. Bioinformatics, 2011, vol.27(3):pp.326-333.

[11] Gerlt,J.A., Babbitt,P.C. and Rayment,I. Divergent evolution in the enolase superfamily: the interplay of mechanism and specificity[J]. Arch.Biochem.Biophys, 2005, vol.433(1):pp.59–70.

[12] Hazel M. et al. The Crotonase Superfamily: Divergently Related Enzymes That Catalyze Different Reactions Involving Acyl Coenzyme A Thioesters[J]. Acc. Chem. Res., 2001, Vol.34(2):pp.145-157.

[13] Seibert,C.M. and Raushel,F.M. Structural and catalytic diversity within the amidohydrolase superfamily. Biochemistry, 2005, vol.44(17):pp.6383–6391.

[14] Pegg.S.C.H. and Shoshana,D.B. et al. Leveraging enzyme structure-function relationship for functional inference and experimental design: the Structure-Function Linkage Database[J]. Biochemistry, 2006, vol.45(8):pp.2545-2555.

[15] Shannon,P. and Markiel,A. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks[J]. Genome.Res., 2003, vol.13(11):pp.2498–2504.

[16] Van Dongen,S. Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht, May 2000.