Performance Analysis of Molecular Complex Detection in Social Network Datasets

Neny Sulistianingsih
Dept. of Informatics, STMIK Bumigora, Mataram
Dept. of Computer Science and Electronics, Faculty
of Mathematics and Natural Sciences, Universitas
Gadjah Mada, Yogyakarta

Edi Winarko
Dept. of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences,
Universitas Gadjah Mada, Yogyakarta

ABSTRACT

Researches related graph dataset conducted for years. One of its main topics was community detection. The development of algorithms to do community detection continuously conducted by adjusting characteristics of datasets used. One of which is Molecular Complex Detection (MCODE) algorithm used to community detection in a dataset of protein-protein interaction (PPI). However, use of the algorithm still limited to PPI dataset only. The aim of this research was to conducted experiment usage of MCODE algorithm in other datasets such us social network datasets. An experiment conducted by comparing the performance of MCODE with other benchmark algorithms such us Label Propagation and Girvan-Newman. From the experiment performed was resulted that for modularity MCODE showed the best result when compared with others, followed Girvan-Newman and Label Propagation with its values were 0.67, 0.66 and 0.46, respectively. Furthermore, for a testing parameter such us running time and average clustering coefficient, MCODE showed better result compared with Girvan-Newman and Label Propagation. For running time, MCODE needed mean time as 0.053 s, Girvan-Newman as 0.056 s and Label Propagation as 0.078 s and for test parameter of average clustering coefficient, MCODE was 0.37, Girvan-Newman was 0.44 and Label Propagation was 0.46.

General Terms

Data Mining, Graph Mining, Social Network Analysis, Algorithm

Keywords

Community Detection, Girvan-Newman, Molecular Complex Detection, Label Propagation.

1. INTRODUCTION

In daily life, human will naturally form group or community. The community formation usually based on interest, environment, and interaction between the human. Representation of community formation which is done by human in daily life can be illustrating as community formation in the social network. The user of the social network who has the same interest in the specific area will share posting, followed and interacted with another user that have the same interest with them. From shared, followed, and other interaction will form a community which will group few of user with others based on the interaction between them. Representation of the community formation not only represented by the interaction between users but also in the form of protein-protein interaction (PPI). Both the interaction of users and PPI can be illustrated as a graph.

The graph formed by set of vertex (V) and edges (E), so formally it can be written $G = (V, E)$. Using the structure, social network and PPI illustrated their user and protein interactions. In a social network, users represented as a set of the vertex (V) and interaction between users represent as a set of edge (E). Same as in PPI that represented it proteins as a set of the vertex (V) and interaction of the protein as a set of edge (E). Both social network and PPI included as a complex system because both has a large number of the vertex which constantly interact with each other resulting in enhancement of its edges. An example of a graph representation of social network structure as showed in Fig 1 and PPI as showed in Fig 2.
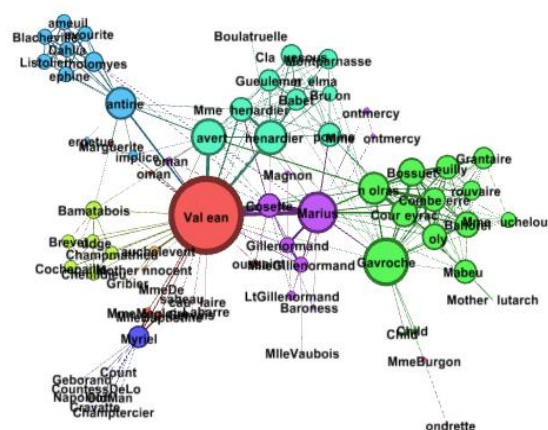

Fig 1: Graph representation of social network structure

The formation of a community that is carried out as a natural trait in real life can also illustrate in the graph that represents it. By utilizing the graph structure, over the past decades, many studies have been conducted to find and develop the most optimal algorithms to detect communities. An algorithm such as Girvan-Newman [1] widely used as benchmark algorithms in this field. Meanwhile, Louvain algorithm based on greedy optimization and Label Propagation algorithms developed based on local community detection performed by unique labeling on each vertex taking into account the maximum labeling of neighbors [2] also used as community detection algorithms. Another algorithm such as Molecular complex detection (MCODE) widely used in community detection of protein-protein interaction dataset [3] and many other algorithms. Some commonly used algorithms for clustering such as K-Means [4] and Fuzzy k-means [5] also used for community detection.

The purpose of this research is to test the use of MCODE algorithm for community detection on social network datasets. This goal achieves by comparing the performance of the

MCODE algorithm with some other benchmark algorithms such as Girvan-Newman and Label Propagation on social network datasets and other datasets. Furthermore, from the community-detection process of each algorithm, the cluster will be addressed to obtain some evaluation parameter values such as modularity, average clustering coefficient, running time and number of clusters formed. Further analysis will be performed on these parameters to determine the best algorithm among the three algorithms used in this study.

2. THEORIES

2.1 Molecular Complex Detection

Since the invention of Molecular Complex Detection (MCODE) algorithm by [6], the algorithm widely used in the field of biology. Some studies like [7]–[9] use the algorithm on both the PPI dataset, MRI and network activity. The efficiency of MCODE algorithm [9] regarding community detection is one of the reasons for a large number of studies in the field of biology using the algorithm.

The Molecular Complex Detection (MCODE) algorithm is a clustering algorithm based on the density calculation of a graph [9]. Using this concept, MCODE performs k-core calculations, that is, the calculation of minimum degree values in a graph. The basis for the use of k-core concepts is that on each graph nodes found in the graph must be connected to other nodes, so the highest k-core values represent an area of the graph that has the most solid connectivity.

The MCODE algorithm consists of three steps such as:

1) Scoring

This step conducted by assigning a score to a node in a neighbor which has been more interconnected. The process of assigning a score starting with the search for the highest k-core value on the node. Furthermore, from the node will be calculated core density. Assigning a score from nodes done by the formula:

$$Score = K \times Core\ Density \qquad (1)$$

The process repeated until all nodes have scored. Pseudo code for this step is as follows [9].

Require:
 $G = (V, E)$ - graph
  threshold – vertex weight

Ensure:
  Set of clusters
1: **for** all vertex v in G do
2:    N = the local graph consisted of v and neighbors; K=sub-graph on N which has highest k-core;
3:    k=k-core of K:
4:    density=density of K;
5:    set weight of v to k*density
6:
7: **end for**

2) Cluster Finding

At this step, the search for a node to be seed. The criterion of a node to be seed is the node that has the highest value. Furthermore, from the seed, the cluster is formed based on the nodes around the seed that passes the threshold value. This process repeated so that all clusters identified. The threshold calculation is performed using the formula:

$$Threshold = (1.0 - Node\ Score\ Cutoff) \times Score\ of\ Seed\ Node \qquad (2)$$

The value of node score cutoff is a parameter value inputted by the user with a range of values from 0-1. The greater the value of the given node cutoff score, the more nodes that enter into a cluster causes the larger the cluster formed. Pseudo code for the cluster finding the step to find the seed in the MCODE algorithm is as follows [9].

Require:
 $G = (V, E)$ - graph
  threshold – vertex weight
Ensure:
  Set of clusters
1: **for** all vertex v in G do
2:    **if** v is not null **then**
3:      seed=the vertex which has the maximum weight in the graph G;
4:      Call: Find_cluster(G, seed,threshold);
5:    **end if**
6: **end for**

From the pseudo code to find the seed, then there is the pseudo code to find the cluster on the MCODE algorithm [9].

Require:
 $G = (V, E)$ - graph
  seed
  threshold – vertex weight
Ensure:
  Set of clusters
1: **if** s already seen **then**
2:    **Return**
3: **end if**
4: **for** each vertex v neighbors of s **do**
5:    **if the** weight of v=(weight of s)*(1-threshold) **then**
6:      add v to cluster containing s
7:      call Find_cluster(G,v,threshold)
8:    **end if**
9: **end for**

3) Post Processing (Optional)

At this step, the process processed on nodes that are singly-connected. There are two options of the method, namely removing or adding the node to a cluster. The process of removing singly-connected nodes is called Haircut. For the process of adding a node to a cluster, the process done by considering the given Density Cutoff. The process addition of nodes to a cluster is called Fluff.

2.2 Girvan Newman

The invention of the Girvan-Newman algorithm by [1] has become one of the triggers of research development on the topic of community detection for graph dataset. Girvan-Newman especially has been used in further studies, not only using the basic concepts as performed by [10]–[12] but also the development of the algorithm, as the research done by [13], [14]. Until now the development of this algorithm continues to be done considering many possible developments related to the concept of the Girvan-Newman algorithm.

The main concept of the Girvan-Newman algorithm is edge betweenness. The concept primarily emphasizes the most passed edge concepts by all vertices with shortest paths. From the edge that has the highest value of edge betweenness, then

carried out the removal of the edge repeatedly to form a cluster. The steps in the Girvan-Newman algorithm are as follows.

1) Edge-Betweenness Calculation

At this step, we calculate the edge betweenness value of all edges of the graph used. The calculation of edge betweenness value done by using Breadth First Search (BFS) approach. The use of BFS to calculate the edge betweenness is done to determine the number of shortest paths from the source to each node and the distance from the source assigned to each node (node marking). Furthermore, the number of shortest paths through the edge is calculated for each edge starting from the last node visited from the use of BFS.

2) Removal edge with the highest edge betweenness value

The removal of the edge begins at the edge that has the highest edge betweenness value in the graph used. However, if there is the same value of highest edge betweenness in a graph, then the selection for edge removal will be made randomly.

3) Re-calculation of edge betweenness value

At this step, after the edge with the highest edge betweenness value is removed, then re-calculation of edge betweenness value at each edge.

4) Back to step 2

2.3 Label Propagation

Like the Girvan-Newman algorithm, Label Propagation algorithm is one of the benchmark algorithms on detection community for graph datasets. Since it is developed by [2], the use of the algorithms is quite popular. The reason is mainly that of its use which does not require any parameters, easy implementation, fast execution for large networks, and the ability to detect valid communities in random graphs [15]. For that reason, studies using the algorithms continue, as conducted by [16]–[20].

In contrast to the concept of edge betweenness used in Girvan-Newman, Label Propagation algorithm uses its graph structure. The Label Propagation algorithm performed by labeling each node with a unique label and sequence for labeling process is random. The labeling based on the majority of labels owned by a neighbor of nodes to be labeled. Furthermore, the cluster will be formed based on the similarity of the label owned by the nodes. The steps in the Label Propagation algorithm [2] are as follows.

1) Label initialization to all nodes in graph. For a node x given $C_x(0) = x$.
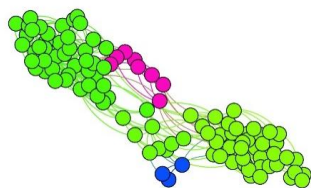
2) Set $t = 1$

3) Set nodes in graph with random order and set to X

4) For each $x \in X$ chosen in specific order, then $C_x(t) = f(C_{xi1}(t), \dots, C_{xim}(t), C_{xi(m+1)}(t-1), \dots C_{xik}(t-1))$. f here returns the highest-frequency labels between neighbors and broken relationships uniformly and randomly.

5) If each node already has a label with a maximum value of the number of neighbors they have, then the algorithm stops. If not, then set and start from step (3).

3. RESEARCH METHODOLOGY

3.1 Datasets

The dataset used in this study consisted of several datasets such as dataset of the weighted network of characters in the novel Les Miserables [21] (Les Miserables), social network of friendships between 34 members of a karate club at a US university in the 1970s [22] (Zachary's Karate Club) and a social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand [23] (Dolphins). These datasets included in the social network dataset. In addition to these three datasets, two additional datasets used, namely the political book network dataset in the United States during the 2004 presidential election and sold by Amazon.com [24] (Polbooks) and datasets of adjacency network of adjective and noun words on David Copperfield's novel by Charles Dickens [25] (AdjNoun). The graph used in this paper is undirected graph even though there are datasets can be used as a directed or undirected graph. The tools used in this study include Gephi 0.8.2 and Cytoscape 3.5.1. Gephi is used to test Label Propagation algorithm while Cytoscape used for MCODE and Girvan-Newman algorithms. For the Girvan-Newman algorithm, the Cytoscape tool represented with GLAY plugins, a plugin used for community detection based on the Girvan-Newman algorithm. Information on the dataset used as showed in Table 1.

Table 1. Datasets Information

No.	Dataset	# of Nodes	# of Edges	Types of Graphs
1	Zachary Karate Club	34	78	Undirected/Directed
2	Les Misarable	77	254	Undirected/Directed
3	Polbooks	105	441	Undirected
4	Adjnoun	112	425	Undirected
5	Dolphins	62	159	Undirected

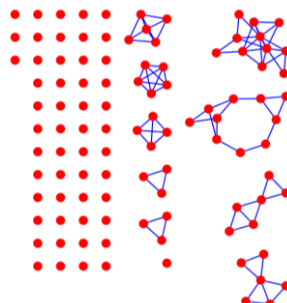**Fig 3: Visualization of cluster from PolBooks dataset with Label Propagation**

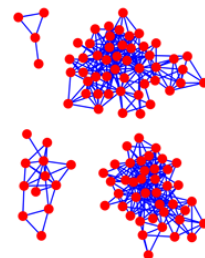**Fig 4: Visualization of cluster from PolBooks dataset with MCODE**

Fig 5: Visualization of cluster from PolBooks dataset with Girvan-Newman

3.2 Experiment

The test will be performed using five data sets, consisting of three social network datasets and two other datasets. Each of these datasets will test with all three algorithms, Label Propagation, MCODE, and Girvan-Newman. The three algorithms will use the standard parameters used in each algorithm, especially in the test using MCODE will be done by using Node Score Cutoff value of 0.2, k-core of 2 and Haircut selected as Post Processing method. Furthermore, from the cluster results of each algorithm will be analyzed several evaluation parameters, also, to seeing the quality of clusters resulted. The evaluation parameters to be analyzed in this research are as follows.

1) Modularity

Modularity is used to measure the quality of the resulting cluster [26]. In this study will be used global modularity of the resulting cluster. The value of modularity ranges from 0-1 with a value of modularity close to 1 indicates better cluster quality.

2) Average Clustering Coefficient

Average Clustering Coefficient is used to indicate the proximity relationship between neighbors of a node [9]. The Average Clustering Coefficient value ranges from 0-1 with an Average Clustering Coefficient value close to 0 indicating closer proximity between neighbors.

3) Running Time

Running time is the length of computation time required to complete the clustering process. Running time measurement is done by using seconds units.

4. RESULT

Tests conducted on five datasets in resulting several clusters according to each of algorithms. The number of clusters generated depends on the number of nodes and edges that the datasets have respectively. Visualization of clusters generated from algorithms of Label Propagation, MCODE and Girvan-Newman can be seen in Figure 3, 4 and 5.

In the use of Label Propagation algorithm, the mean of cluster numbers produced is 4.40. While on the MCODE algorithm, the mean of cluster numbers generated is 5.00 and the Girvan-Newman algorithm is 4.60. From the results was known that Label propagation algorithm is the algorithm that produces the smallest number of clusters when compared to Girvan-Newman and MCODE algorithm. Comparison of the number of clusters produced by the three algorithms as shown in Fig 6.
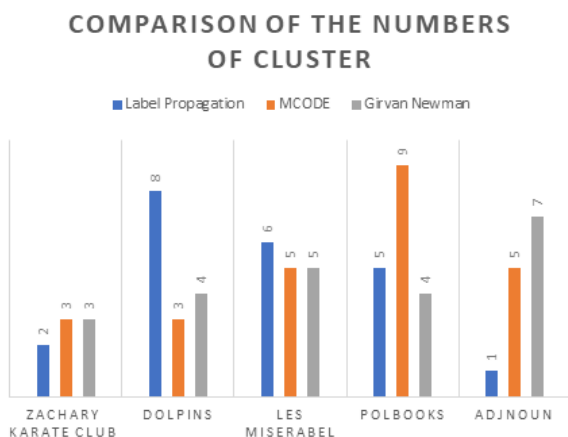

Fig 6: Comparison of the number of clusters resulted

Furthermore, from the cluster results are tested the modularity of each algorithm. From the results of the tests performed, the mean modularity of Label propagation is 0.46 while MCODE and Girvan-Newman are 0.67 and 0.66, respectively. From the analysis known that the MCODE algorithm has the highest value of modularity, followed by Girvan-Newman and Label Propagation, respectively. Although the difference in mean modularity between MCODE and Girvan-Newman is not significant, that is, only 0.01, this shows that MCODE resulted better cluster compare with Girvan-Newman and Label Propagation. The comparison of modularity values of all three algorithms as showed in Figure 7.
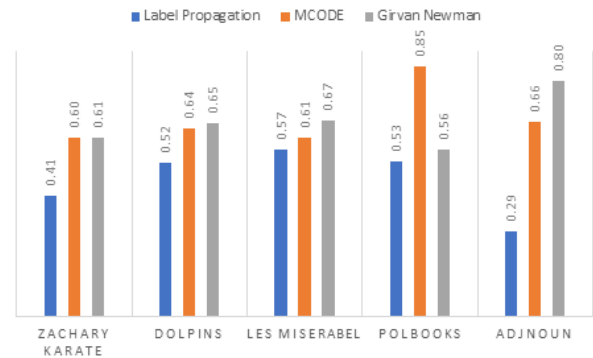

Fig 7: Comparison of modularity

The analysis of Average Clustering Coefficient value from the results cluster from three algorithms shows that the means of Average. Average Cluster Coefficient for Label Propagation algorithm resulted is 0.46, for MCODE and Girvan-Newman are 0.37 and 0.44, respectively. From these results note that the MCODE algorithm has the highest value of Average Clustering Coefficient, followed by Girvan-Newman and Label Propagation algorithm, respectively. The comparison of the Average Clustering Coefficient value of the three algorithms as showed in Figure 8.
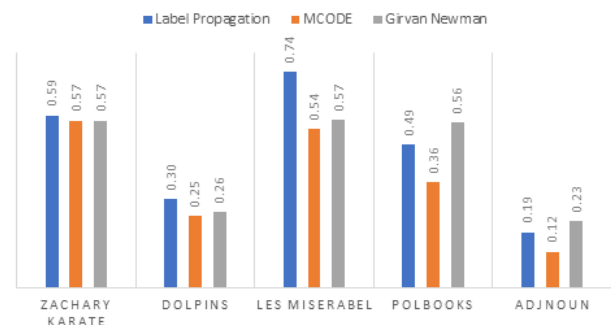

Fig 8: Comparison of average clustering coefficient

Regarding computation time of each algorithm, the mean running time for Label Propagation algorithm is 0.078 s, while for MCODE is 0.053 s and Girvan-Newman is 0.056 s. The result shows that for the computation time of each algorithm, the fastest is MCODE, followed by Girvan-Newman and Label Propagation respectively. The comparison of the mean running time of the three algorithms as showed in Figure 9.
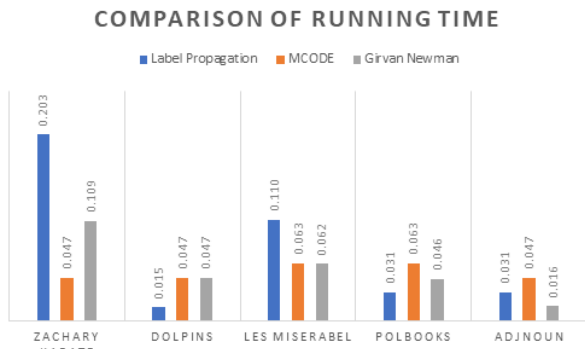
Fig 9: Comparison of running time

If the datasets grouped according to its type, there are some changes that significantly affect the ranking of the three algorithms. In the social network dataset, Girvan-Newman has the highest modularity value of 0.642 followed by MCODE and Label Propagation of 0.616 and 0.498 respectively. As for other datasets, MCODE algorithm has the highest modularity value of 0.753, followed by Girvan-Newman and Label Propagation algorithms of 0.680 and 0.407 respectively. For the means value of Average Clustering Coefficient on both a social network and other datasets, MCODE has better results than Girvan-Newman and Label Propagation algorithms. In the social network dataset, the means value of MCODE Average Clustering Coefficient is 0.452, and for other datasets is 0.242. As for Girvan-Newman algorithm is 0.468 for social network dataset and 0.396 for the other dataset. For Label propagation, the mean value of Average Clustering Coefficient for the social network dataset is 0.542 and 0.339 for the other dataset. Related to the running time of each algorithm, MCODE shows better results for running time on the social network dataset. MCODE has a mean running time of 0.052 for the social network dataset. Faster than running time on Girvan-Newman and Label Propagation algorithms on the social network dataset which is 0.073 and 0.109, respectively. As for other datasets, MCODE is longer than Label Propagation and Girvan-Newman algorithm, which is 0.055 when compared to the Label Propagation and Girvan-Newman running time which equals to 0.031. Comparison of three algorithms in social network datasets as showed in Figure 10 and in other datasets as showed in Figure 11.
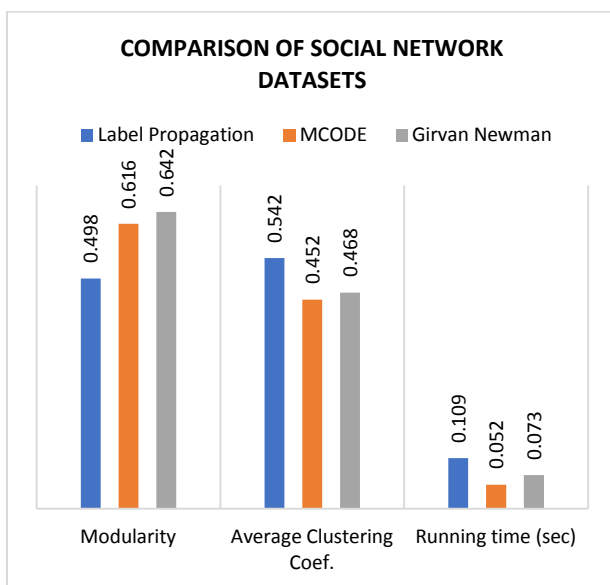

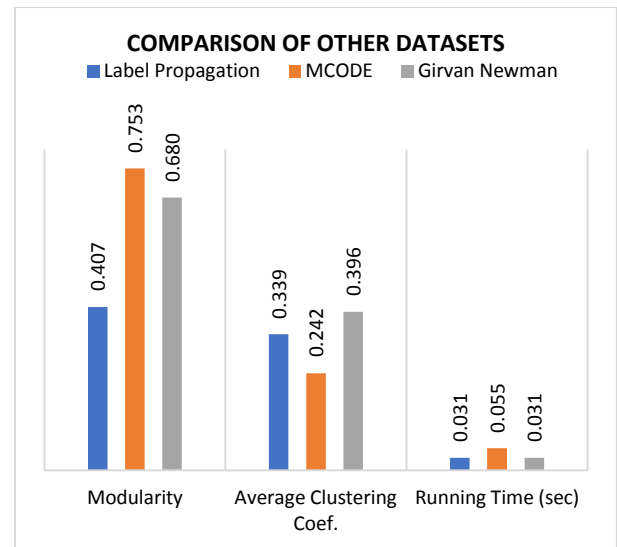**Fig 10: Comparison of social network datasets**

Fig 11: Comparison of other datasets

Further analysis of the effect of the number of nodes and edges of each graph also tested in this study. However, the results of the research found no effect of the number of nodes and edges against the values of modularity, Average Clustering Coefficient and the number of clusters resulted. However, the number of nodes and edges affects only the running time of the algorithm, especially in the MCODE algorithm.

5. CONCLUSION

From the results of the analysis of the five datasets used in this study known that on the mean value of the parameters of the number of clusters, modularity, Average Clustering Coefficient and running time resulted, MCODE algorithm is superior to Girvan-Newman and Label Propagation algorithm. However, the superiority of MCODE is not very significant when compared with the results of community detection using the Girvan-Newman algorithm. Even the difference between these two algorithms is only 0.01 for the mean value of modularity. As for the parameters of Average Clustering Coefficient and running time, the difference is only 0.07 and 0.03, respectively. Also, the social dataset of MCODE algorithm also shows good results, both regarding modularity, Average Clustering Coefficient and running time. The test shows promising results for the use of MCODE algorithms on datasets outside the PPI datasets.

For further research, there is still much research that can conduct for the development of MCODE. The result of modularity showing the quality of clusters produced still shows a not-so-high number ranging from 0.60 to 0.85 for the MCODE algorithm, 0.29 to 0.57 for Label Propagation algorithms and 0.56 to 0.80 for the Girvan-Newman algorithm. The result proves that the development of these three algorithms is still very possible. Especially the MCODE algorithm because the computational time of the MCODE is fast enough compared to other algorithms.

Also, further research on the factors that influence the cluster results of the MCODE algorithm is necessary. With the prior research related factors that influence the result clusters in MCODE algorithm, further research to develop MCODE algorithm can focus on the factor that has the greatest influence on the MCODE algorithm. From these developments, it can be obtained better cluster results from MCODE algorithm especially for datasets other than PPI.

6. REFERENCES

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 12, pp. 7821–6, 2002.

[2] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 76, no. 3, pp. 1–12, 2007.

[3] A. Clauset, M. E. J. Newman, and C. Moore, "2004_Clauset_Finding community structure in very large networks.pdf." Pysichal review, pp. 1–6, 2004.

[4] J. Macqueen, "Some methods for classification and analysis of multivariate observations," *Proc. Fifth Berkeley Symp. Math. Stat. Probab.*, vol. 1, no. 233, pp. 281–297, 1967.

[5] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2–3, pp. 191–203, 1984.

[6] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 1, p. 2, 2003.

[7] P. O'Driscoll, E. Merényi, C. Karmonik, and R. Grossman, "SOM and MCODE methods of defining functional clusters in MRI of the brain," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2014, pp. 734–737, 2014.

[8] S. Brohée and J. van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks.," *BMC Bioinformatics*, vol. 7, p. 488, 2006.

[9] J. Zhong, L. Liu, Y. Wei, D. Luo, L. Sun, and Y. Lu, "Personalized activity recognition using molecular complex detection clustering," *Ubiquitous Intell. Comput. 2014 IEEE 11th Intl Conf IEEE 11th Intl Conf Auton. Trust. Comput. IEEE 14th Intl Conf Scalable Comput. Commun. Its Assoc. Work.*, pp. 850–854, 2014.

[10] M. E. J. Newman, "Detecting community structure in networks," *Eur. Phys. J. B - Condens. Matter Complex Syst.*, vol. 38, no. 2, pp. 321–330, 2004.

[11] T. Matijevi and T. Vujičić, "Performance Analysis of Girvan-Newman Algorithm on Different Types of Random Graphs," pp. 11–16, 2016.

[12] D. Yanrui, Z. Zhen, W. Wenchao, and C. Yujie, "Identifying the Communities in the Metabolic Network Using 'Component' Definition and Girvan-Newman Algorithm," *2015 14th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci.*, pp. 42–45, 2015.

[13] B. Kong, L. Zhou, and W. Liu, "Improved modularity based on girvan-newman modularity," *Proc. - 2012 Int. Conf. Intell. Syst. Des. Eng. Appl. ISDEA 2012*, pp. 293–296, 2012.

[14] L. Despalatović, T. Vojković, and D. Vukičević, "Community structure in networks: Girvan-Newman algorithm improvement," *2014 37th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2014 - Proc.*, no. May, pp. 997–1002, 2014.

[15] A. Lakhdari, A. Chorana, H. Cherroun, and A. Rezgui, "A link strength based label propagation algorithm for community detection," *Proc. - 2016 IEEE Int. Conf. Big Data Cloud Comput. BDCloud 2016, Soc. Comput. Networking, Soc. 2016 Sustain. Comput. Commun. Sustain. 2016*, pp. 362–369, 2016.

[16] Z.-W. Liang, J.-P. Li, F. Yang, and A. Petropulu, "Detecting community structure using label propagation with consensus weight in complex network," *Chinese Phys. B*, vol. 23, no. 9, p. 98902, 2014.

[17] A. K. K. C., L. Jacques, and C. De Vleeschouwer, "Discriminative and Efficient Label Propagation on Complementary Graphs for Multi-Object Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8828, no. c, pp. 1–1, 2015.

[18] Y. He, J. Xu, and B. Yuan, "Community Structure Analysis Using Label Propagation and Flow-Based Ensemble Learning," pp. 720–726, 2016.

[19] T. Swearingen and A. Ross, "Predicting Missing Demographic Information in Biometric Records using Label Propagation Techniques," 2016.

[20] G. Liu, K. Meng, H. Guo, L. Pan, and J. Li, "Automatic Threshold Calculation Based Label Propagation Algorithm for Overlapping Community," *2016 IEEE First Int. Conf. Data Sci. Cybersp.*, pp. 382–387, 2016.

[21] D. E. Knuth, *The Stanford GraphBase : a platform for combinatorial computing*. Boston, 2009.

[22] W. W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, 1977.

[23] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, Sep. 2003.

[24] V. Krebs, "Political Books -- Polarized Readers -- May 2004," 2004. [Online]. Available: http://www.orgnet.com/divided3.html. [Accessed: 05-Jun-2017].

[25] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 74, no. 3, 2006.

[26] M. Chen and B. K. Szymanski, "Fuzzy Overlapping Community Quality Metrics," *Soc. Netw. Anal. Min.*, vol. 540, 2015.