Community Detection Algorithm for Finding Overlapping and Hierarchical Structure

Yunku Yeu[1], Jaegyoon Ahn[2] and Sanghyun Park[1*]

[1]Department of Computer Science, Yonsei University
C533-1 Engineering Hall III, Shinchon-Dong, Seodaemoon-Ku, Seoul, Korea
[2]Department of Integrative Biology and Physiology, University of California, Los Angeles
611 Charles E. Young Drive, Room 660, Molecular Biology Institute, UCLA Los Angeles, CA 90095
yyk@cs.yonsei.ac.kr, jgahn@ucla.edu and sanghyun@cs.yonsei.ac.kr

Community finding is an important step to understand huge and complex network. In real world, communities may have overlaps and hierarchical structure among them. Here we propose a novel clustering algorithm that is capable of finds communities considering those characteristics. Proposed method detects overlapping communities without limitation of shape and position. We applied our method on yeast PPI (Protein-Protein Interaction) network with current methods. As a result, our method produce most balanced result in terms of size distribution of cluster and overlap. Also, our method finds clustering results that are significantly enriched by known biological knowledge.

Key Words: community detection, clustering, overlapping community, hierarchical structure

1. INTRODUCTION

Network is a useful tool to analyze large and complex data. In network, an object is represented as a node, and an interaction between the objects is represented as an edge. The dynamics of interactions can be represented as weights of edges. An important advantage of network is ease of integration. Complex and different data can be reduced by few simple concepts – nodes, edges, weights and direction of the edges. Network is very useful when large amount of data are created by various data-sources, such as biological domain. [1]

Researchers can infer hidden meanings from the network by analyzing its characteristics. One of basic analyzing processes is finding community structure. Although there is no common definition of community, it is widely accepted that a community or a cluster should have larger intra-similarity than inter-similarity.

Another important assumption about community is that nodes from same community are likely to have similar characteristics. We can infer characteristics of a community based on few well-known member nodes. Missing links inside of dense-connected community can be predicted. In pharmacy, we can predict side-effects by observing target protein's community.

In real world, community structure is more complicate. For example, an object or an individual can belongs to many groups therefore some overlaps are arisen between those communities. CPM (Clique Percolation Method) [2] is one of the popular algorithms finding overlapping communities. Firstly, this method finds k-cliques from the network then unions of adjacent k-cliques are reported. The nodes belong to more than one union represent overlaps. Other method [3] defines community as a local optimization of given function. They find many local optimal

* Corresponding author

subsets. Nodes can be visited by many subsets, and those nodes represent the overlaps.

Furthermore, many communities or individuals have hierarchical structure in real world. It is a natural fashion that small communities are grouped together to form a larger community. In biological domain, for example, huge amount of biological knowledge is organized into Gene Ontology (GO) database [4]. In GO database, all knowledge is divided into three largest categories – biological process, cellular component and molecular function. And each category is divided into sub-categories again. Hierarchical clustering and multi-resolution method [5] are popular for finding hierarchical structure. Multi-resolution method has freely tunable parameters to controls the size of clusters. This method shows the hierarchical structure by presenting many partitions with different parameters.

There are some researches to find above two features simultaneously. EAGLE [6] finds maximal cliques and applies hierarchical clustering among them. Nodes that are included in different maximal cliques can be overlapped. Lancichinetti et al [7] find a natural community for all nodes. A natural community starts from a single node, and expands to its neighbor to maximize a fitness function. Nodes can be visited by many natural communities. A random set of natural communities which can cover entire network are chosen as a clustering result. Ahn et al [8] propose edge-based hierarchical clustering method. A node can be overlapped corresponds to its degree.

The short points of current works can be summarized as follows: The first short point is that only nodes at the border of communities can be overlapped. Ahn, CPM, EAGEL have this short point. Second short point is that they have limitation about the shape of community. Because network may have some missing link, they can miss some communities if they can find only cliques. CPM and EAGLE have this problem. The non-deterministic feature is the last short point. Lancichinetti's method has this problem.

In this article, we propose a clustering method which has no limitation about shape and position of overlap. Our method is a transformation of agglomerative hierarchical clustering. When we merge most similar pair, we also find adjacent nearly-similar pair and merge them at the same time. Common nodes that are included in those pairs could be overlapped. We applied our algorithm with other methods - CPM and Ahn's method - to Protein-Protein Interaction (PPI) network of *Saccharomyces cerevisiae*(baker's yeast). Our algorithm finds feasible size of overlaps. Communities and overlaps found by our method are significantly enriched by GO database.

This paper is organized as follows: The second section shows detail of our idea. The third section shows descriptions about experimental setting. In the fourth section, we report experimental results and analysis. In the last section, the conclusion is presented.

2. METHOD

2.1 Basic idea

Our algorithm is based on the agglomerative hierarchical clustering algorithm. When we merge most similar pair (best pair), we also find 'nearly similar' pairs among best pair's neighbor. A parameter *T* determines whether a pair is 'nearly similar' or not. If more than one pairs are found as nearly similar, they produce overlapping communities. In figure 1, For example, edge weights represent similarity between nodes and (*A*, *B*) is the best similar pair. Because parameter *T* is set to 0.8, any similarity value greater than 0.72 (= 0.9 x 0.8) is considered as nearly similar. Thus, pair (*A*, *C*) is merged at the same time, new community *AB* and *AC* are constructed. At this point, common node A is belongs to adjacent two communities, so these two communities have an overlap. Right side of figure 1 shows this merging step with dendrogram. The lowest nodes in the dendrogram, (A, B) and (B, C) are grouped simultaneously.

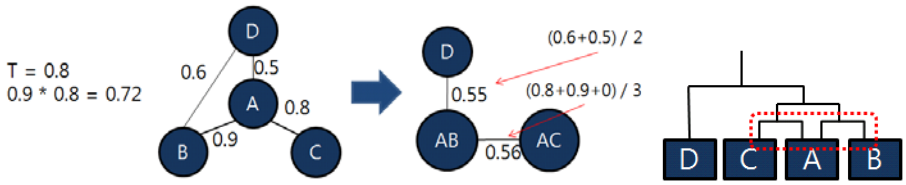

Figure 1. Example of merging process

In the cases such as figure 2, if all neighbors of best pair (*A*, *B*) are nearly similar, the number of nodes in the network can be increased permanently. We can't guarantee deterministic running time of our algorithm in this situation. In order to prevent infinite running, we skip checking the nearly-similarity when target node is common neighbor of current merging nodes. Node *C* and *D* in figure 2, Node *D* in figure 1 are in this case.

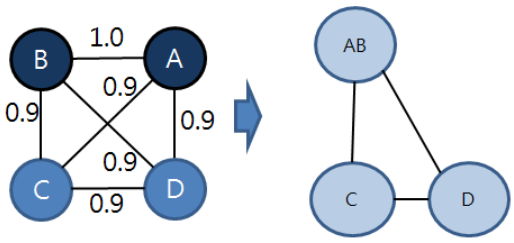

Figure 2. Exclusion of common neighbor in merging step

Our method might merge a node with many other nodes, network topology and similarity values should be updated after each iteration. Let assumes that n_i represents node *i*, $S(n_i, n_j)$ is the similarity value between node n_i and n_j (if n_i and n_j have no connection, $S(n_i, n_j) = 0$), N_i is set of neighbor of node n_i. For each iteration, similarity values are updated as follows:

1) When a node pair (n_i, n_j) are merged into a new node n_{ij},

$$\text{for all } n_x \text{ in } \{N_i \cup N_j\}, \qquad S(n_x, n_{ij}) = \frac{1}{2} \times (S(n_x, n_i) + S(n_x, n_j)) \tag{1}$$

2) If two adjacent pairs (n_i, n_j) and (n_i, n_k) are merged simultaneously and construct two nodes n_{ij} and n_{ik},

$$S(n_{ij}, n_{ik}) = \frac{1}{3} \times (S(n_i, n_j) + S(n_i, n_k) + S(n_j, n_k)) \tag{2}$$

Our method have a time complexity of $O(n^3)$. (n = number of nodes) This complexity is larger than traditional node-based hierarchical clustering ($O(n^2)$), but smaller than edge-based hierarchical clustering ($O(n^4)$).

2.2 Partition density

In order to produces hierarchical clustering result, a cut position in the dendrogram must be determined. We exploited the partition density [9] as a quality function, and determined cut position where this function is maximized. Partition density D is calculated as follows :

$$\mathrm{D} = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \tag{3}$$

where M represents number of edges in the network, c is a cluster, n_c and m_c represent the number of nodes and edges in the cluster c respectively.

3. EXPERIMENTAL SETTING

In order to evaluate the performance of our algorithm, we compared our method with two current algorithms. The first algorithm is CFinder [9] based on CPM. The second algorithm is Ahn's edge-based hierarchical clustering.

3.1 Datasets

In this paper, we used PPI data of *Saccharomyces cerevisiae* for performance comparison. We downloaded yeast PPI dataset from CCSB [10]. Table 1 shows the description of these datasets.

Table 1. Description of CCSB dataset

# of nodes	# of edges	Average degree
1,275	13,079	10.25

3.2 Weighting scheme

Because our method is a sort of hierarchical clustering, we can calculate weights for edges freely. In this experiment, we exploited topology-based weighting scheme. Our method and CFinder need similarity value between node-node, so equation (4) is used. For Ahn's method, similarity values between edge-edge are calculated using equation (5).

$$S(n_i, n_j) = \frac{|N_{+i} \cap N_{+j}|}{|N_{+i} \cup N_{+j}|} \quad (4)$$

$$S(e_{ik}, e_{jk}) = \frac{|N_{+i} \cap N_{+j}|}{|N_{+i} \cup N_{+j}|} \quad (5)$$

In above equations N_{+i} means the union of node n_i and its neighbor.

3.3 Evaluate clustering quality

The clustering results are compared according to following three criterions. The first criterion is the size distribution of clusters. If a clustering result contains a few but huge clusters that cover most fraction of entire network, it won't be a good clustering result in many cases. The second criterion is coverage of clustering results. Good clustering results should contain as many as nodes in the network while maintaining clustering quality. The last one is that how clustering results are significantly enriched by biological knowledge (i.e. GO database). We used Funcassociate [11] for this test. Significantly enriched set of genes (proteins) could be regarded as involving similar biological functions. The last criterion means degree of association between clusters and biological functions.

3.4 Evaluate overlap quality

The qualities of overlaps are also compared according to following two criterions. The first one is statistics about overlap size. Although algorithm finds many communities and overlaps, it is better to detect a single larger community then almost overlapped small communities. The second criterion is GO enrichment test for overlap area. Although this is not true in every cases, overlap area can be involved some biological functions itself.

4. EXPERIMENTAL RESULTS

The system used for performance comparison includes Intel I3 3.07GHz CPU and 4GB memory. Three different algorithms are evaluated. Table 2 shows running time of each algorithm. Proposed method showed better performance than edge based hierarchical clustering algorithm.

Table 1. Running time of each algorithm

Our method (OH)	Ahn et al	CFinder
2m 36s	33m 8s	12s

4.1 Cluster quality

Table 3 shows an analysis about clustering results. Proposed method found smaller number of clusters than Ahn's method. Also, our method showed smaller coverage than other two methods. However, our method found clusters with more evenly distributed size. The largest cluster found by Ahn's method contains 443 nodes. It

corresponds to 35% of entire nodes in the network, and 50% of nodes in the clustering result. Also, CFinder assigned 488 nodes into a largest cluster. This number corresponds to 90% of nodes in the clustering result. Furthermore, average size of largest x% clusters is decreased more rapidly in Ahn and CFinder. When we consider only largest 10% clusters, average size of them are 16.75, 96.17 and 127.75 respectively. If we consider largest 20% clusters, the average size are decreased to 13.81, 56.33 and 75.71.

Table 2. Size distribution of found clusters

Items	Our method(OH)	Ahn et al	CFinder
Number of clusters	76	117	35
Average size of clusters	7.58	17.29	19.28
Maximum size of clusters	23	443	488
Median size of clusters	6	7	5
Average size of top 10% large clusters	16.75	96.17	127.75
Average size of top 20% large clusters	13.81	56.33	75.71
Number of nodes covered by any cluster	448 (35.14%)	867 (65%)	547 (42.9%)

Figure 3 shows the fraction of significantly enriched clusters by GO. 90.7% of our clusters were enriched with significance level $a < 0.05$.

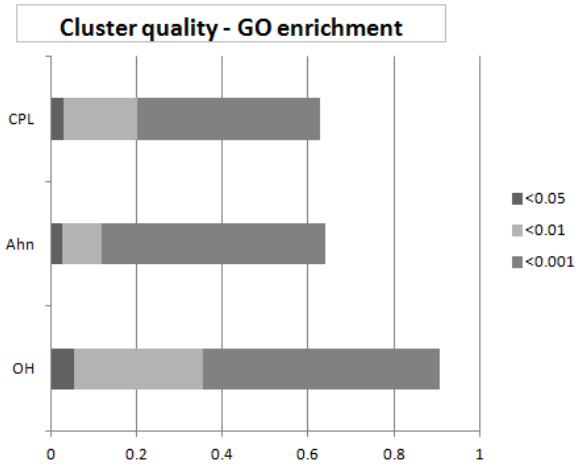

Figure 3. Fraction of clusters that are significantly enriched by GO database

4.2 Overlap quality

Table 4 shows a statistical analysis about found overlaps. Ahn's method founds overlaps per all adjacent two clusters due to its algorithmic characteristics. Thus, Ahn's method found much more overlaps compared with the number of clusters. Furthermore, Ahn's method found 867 nodes in clustering result and 466 nodes of them placed in overlap area. In other word, a half of the clustering result is the overlap area.

Table 3. Overlap quality of clustering

Items	Our method (OH)	Ahn et al	CFinder
# of clusters	76	117	35
# of overlap areas	30	371	29
# of nodes in clusters	448	867	547
# of nodes in overlaps	92 (20.5%)	466 (53.7%)	99 (18.0%)

Figure 4 shows fraction of significantly enriched overlaps by GO. 96.6% of our clusters were enriched with significance level $a < 0.05$.

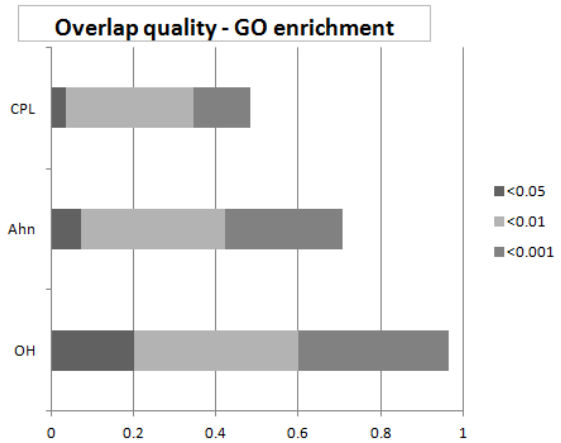

Figure 4. Fraction of overlaps that are significantly enriched by GO database

5. CONCLUSION

This paper proposed a hierarchical clustering algorithm that could find overlapping communities. Proposed algorithm based on simple idea but it can find any shape of overlaps from everywhere in the community. We compared proposed algorithm with two other overlapping communities finding algorithm. Our experimental result shows that our method provided balanced performance in terms of size distribution. Also, our method finds communities that are more significantly involved with biological functions.

As a further study, we will evaluate proposed method with more large and complex network. Test network CCSB is relatively small and sparse network but contains relatively accurate PPI data. In terms of functional quality of clusters, we can assay functional relationships among communities and their overlap. Although they are enriched by GO terms independently, those GO terms should have close relationship.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (2012R1A2A1A01010775).

REFERENCES

[1] Lee, I.-S., Lehner, B., Crombie, C., Wong, W., Fraser, A. G., and Marcotte, E.M, "A single gene network accurately predicts phenotypic effects of gene perturbation in Caenorhabditis elegans," *nature genetics*, vol. 40, no. 2, pp. 181-188, 2008

[2] Palla, G., Derényi, I., Farkas, I., and Vicsek, T., "Uncovering the overlapping community structure of complex networks in nature and society," *nature*, vol. 435, pp. 814-818, 2005

[3] Baumes, J., Goldberg, M., Krishnamoorthy, M., Magdon-Ismail, M., and Preston, N., "Finding communities by clustering a graph into overlapping subgraphs," In *Proc. International Conference on Applied Computing (IADIS 2005)*, Algarve, Portugal, pp. 97-104, Feb, 2005

[4] The Gene Ontology Consortium, "Gene Ontology: tool for the unification of biology," *nature genetics*, vol. 25, pp. 25-29, 2000

[5] Pons, P., and Latapy, M., "Post-processing hierarchical community structures: Quality improvements and multi-scale view," *Theoretical Computer Science*, Vol. 412, No. 8, 2011.

[6] Shen, H., Cheng, X., Cai, K., and Hu, M.-B., "Detect overlapping and hierarchical community structure in networks," *Physica A*, Vol. 388, Issue 8, pp. 1706-1712, 2009

[7] Lancichinetti, A., Fortunato, S., and Kertész, J., "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, Vol. 11, No. 3, 2009

[8] Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S., "Link communities reveal multiscale complexity in networks," *nature*, Vol. 466, pp. 761-765, 2010

[9] Adamcsek, B., Palla, G., Farkas, I. J., Derényi, I., and Vicsek, T., "CFinder: Locating cliques and overlapping modules in biological networks," *Bioinformatics*, Vol. 22, pp. 1021-1023, 2006

[10] http://interactome.dfci.harvard.edu/S_cerevisiae/index.php?page=download

[11] Berriz, G. F., King, O. D., Bryant, B., Sander, C., and Roth, F. P., "Characterizing gene sets with FuncAssociate," *Bioinformatics*, Vol. 19, Issue 18, pp. 2502-2504, 2003