Journal of Engineering and Applied Sciences 12 (11): 2990-2995, 2017
ISSN: 1816-949X

Edge Pruning and GA-Based Clustering Approach for Biological Data Analysis

Athira A. Jolly and Sreeja Ashok
Department of Computer Science and IT, School of Arts and Sciences,
Amrita University, 682 024 Kochi, India

Abstract: Analysis of various kinds of biological data is one of the major problems in bioinformatics. Data mining approaches can be used to uncover hidden patterns and to extract significant knowledge for better analysis and decision making. In this study, we analyse different methods for simplifying the complex networks by identifying significant edges using edge pruning techniques and introduced GA-based clustering process for building optimum subgraphs from the pruned network. The optimum edges were identified by evaluating the similarity between the pair of nodes. Different graph properties like centrality measures are used for positioning the data objects and for improving the cluster cohesiveness. Modularity value was used as the fitness function and mutation operator was performed for deriving the optimum clusters.

Key words: Edge pruning, Genetic algorithm, mutation, centrality, clusters

INTRODUCTION

Complex networks have been used for modeling different entities of the real world, where nodes outline the different objects and edges refers the association between them (Chapela *et al*., 2015). The theory of complex network performs a vital role in variety of major fields, ranging from communication engineering to molecular or population biology (Albert and Barabasi, 2002). Biological system shows complex behavior since they are constructed from microscopic parts which are not directly observable. Hence, any observation regarding their behavior represents a complex function integrating many molecular components and their bonds. It is often too large to visualize and to interpret hidden patterns and meaningful inferences from this integrated network. There are two main challenges when dealing with complex biological networks. First one is to reduce the complexity of the networks and second to partition the data objects into similar groups for in depth analysis. Unsupervised methods are more efficient to really deal with the complex nature of biological data sets. If the complexity can be reduced by transforming the fully connected network to simple ones by maintaining only the most relevant edges and removing irrelevant edges as a preprocessing step before clustering, efficiency can be improved (Ma, 1999).

The mathematical disciplines that emphasize the study of complex network in biology are usually based on graph theory. Understanding the huge complexity of these systems is beyond the classical concept of graph theory. So, we need to incorporate both mathematical tools (probability, dynamical system analysis, matrix analysis and others) and techniques derived from various fields (as statistical mechanics or computer sciences, modern physics) for the extraction of knowledge leading to better decision making. In this study our primary goal is to design a suitable clustering process for complex biological data by simplifying the complex network using edge pruning techniques and building clusters from the pruned network using genetic operators. To acquire higher level optimization, we used mutation operator to find the clusters.

Literature review: There are many researches carried out for network simplification, sub graph extraction, network optimization, etc. The potential application of network analysis using graph theory in the fields of bio-informatics includes identification of drug targets; specifying the role of proteins or genes of unidentified functions, the design of effective control strategies for infectious diseases and the early detection of neural disorders in particular brain regions (Eubank *et al*., 2004). The goal of simplification of network is to remove edges without affecting the flow of network. The methods proposed by Biedl *et al*. (2000) and Misiolek and Chen (2006) are based on connection of the graph and not considering the weights. The path oriented approaches are now achieving more popularity in representing complex co-citation network. In path-oriented approach

Corresponding Author: Athira A. Jolly, Department of Computer Science and IT, School of Arts and Sciences,
Amrita University, 682 024 Kochi, India

proposed by Toivonen *et al.* (2010) edges are deleted while maintaining the best path and the framework proposed was applicable for both instance flow and random graph.

Network epidemiology is widely adopted method for modelling real world disease. To restrict the size of a "network epidemic", Enright and Kitty (2015) suggests that deleting k edges using tree width (an edge deletion to maximal component size at most "h" avoiding "h+1" vertices) will improve the performance. Qasi-clique is another data mining technique for edge pruning proposed by Liu and Limsoon (2008). They simplify the graphs and form sub graphs which represent group of objects sharing some common properties and their pruning method is based on the degree of vertices for identifying the unqualified vertices (Ashok and Judy, 2015, 2016).

The method proposed by Zhou *et al.* (2012) is based on a graph connectivity function for simplifying the graphs for improved visualization of a network. For effective edge pruning they adopt lossy network simplification. For this they consider a weighted graph and a path set. They parameterized their problem with a path quality function q(p) for arbitrary spanning trees, q(p) = 1. Similarly, a connectivity ratio R_k is defined which represents the strength of the graph. It was understood from the study that connectivity cannot increase while removing edges. $R_k = 1$ represents that the pruning of edges will not affect the graphs connectivity. $0<R_k<1$ refers that the pruning of edges affects some loss in connectivity, while $R_k = -\infty$ shows that graph has become a disconnected graph. This approach would help the researchers to explore weighted graphs. Relative neighbourhood graph proposed by Toussaint (1980) uses only relatively close pairs of vertices for simplifying a graph from the distance matrix.

Several community detection algorithms have been discovered using evolutionary algorithms. For complex network, Sumathi-Punitha and Santhanam (2007) adopt Genetic algorithm and perform crossover to find the community structure. They proposed a community detection algorithm; MAGA-NET for optimizing the modularity value for the communities detected using network modularity value as the fitness value and performed cross over operation. Similarly, a multi-agent Genetic algorithm for detecting community in complex network was proposed by Li and Liu (2016) where a series of neighborhood based operators are used for detecting communities. In a survey conducted by Soni and Kumar (2014), various genetic operators are studied on solving NP hard problems and observed that mutation operator is more efficient in detecting meaningful communities.

Pruning algorithms: In the following study, we analysed standard edge pruning algorithms that efficiently identified the significant edges by keeping high level of connectivity. Generally, all algorithms consider a weighted graph G = {V, E} as input. The resultant minor graph is a spanning tree having |V|-1 edges. Thus number of pruned edges can range from 0-|E|-(|V|-1) for all algorithms.

Naive approach: This is the simplest approach where edges are first sorted by their weights in an ascending order. Dijkstra's principle was first used to test whether there exists a shortest path. Then, it repetitively scans the edges from the sorted list and prunes the edge whose removal will not lead to a disconnect graph.

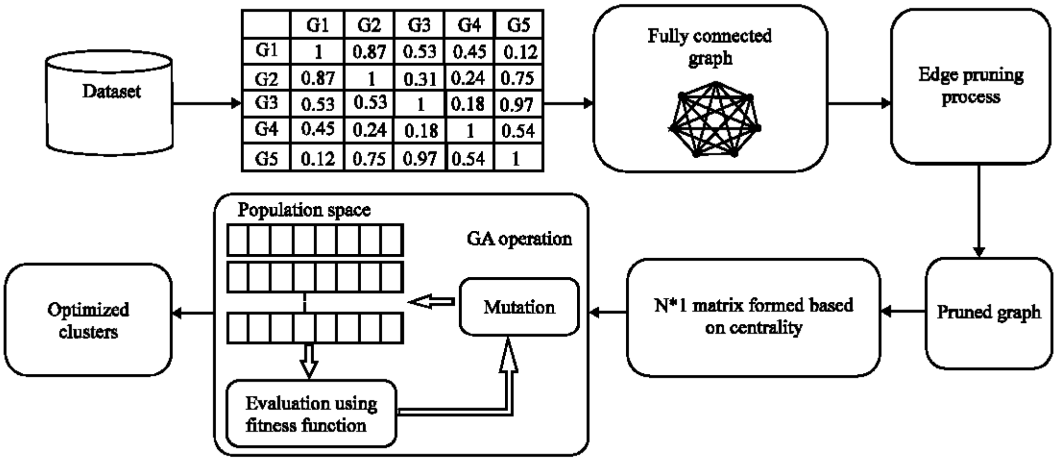
Iterative global algorithm: This algorithm assures that a completely pruned graph is obtained. First it converts a multi graph into a simplified one. Then, it discovers the best possible alternative path P for each edge using Dijkstra's principle (Toivonen *et al.*, 2010).

Brute force approach: This algorithm takes input as path quality function and ratio connectivity and prunes the edges in a greedy style by maintaining the network connectivity. In each iteration for an edge first it calculates the R_k values and deletes the edges based on the maximum R_k value (Zhou, 2012). Integrating these algorithms into the biological network can enhances the performance of the system and thus simply the clustering process.

We used naive approach as a pre-processing step before clustering since this is comparatively simple and it prunes insignificant while maintaining the best path and high level connectivity.

MATERIALS AND METHODS

Proposed system: The proposed clustering process includes two main objectives. In order to simplify the network for reducing the complexity and to improve the performance, naive approach was used to effectively prune the edges while keeping high level connectivity. Later for deriving meaningful partitions, genetic operations are applied on the resultant pruned graph for optimizing the clustering results. Genetic algorithm, an optimization method is used as an active method when the solution space of a random problem is very large and an exhaustive search for the solution is not empirically possible (Bodenhofer, 2003). In Genetic algorithm, the candidate solution members in the solution pool should represent in a right data representation where each



	G1	G2	G3	G4	G5
G1	1	0.87	0.53	0.45	0.12
G2	0.87	1	0.31	0.24	0.75
G3	0.53	0.53	1	0.18	0.97
G4	0.45	0.24	0.18	1	0.54
G5	0.12	0.75	0.97	0.54	1

Fig. 1: Process flow of the proposed algorithm

member in the solution pool is referred as chromosome (Sreeja and Krishnakumar, 2017; Pramanik *et al*., 2010).

Chromosome represents possible solution to the problem and the algorithm finds the best fitting solution member (Leena *et al*., 2016). Genetic algorithm is a faster algorithm for combining a problem into a smaller solution space and if it has a good fitness evaluation function, it produces near optimal solution (Alexandrescu *et al*., 2011; Pramanik *et al*., 2010). We incorporated the centrality feature like degree centrality and closeness centrality for categorizing the data objects in each clusters and the fitness function used is to maximize the modularity of the network.

Pseudo code of the proposed algorithm:

```
Step 1: Similarity computation of N*N matrix
Step 2: Reducing complexity of the network by edge pruning
Step 3: Compute N*1 matrix formed based on closeness/degree centrality
Step 4: Sort the matrix in descending order of magnitude
Step 5: Generate offspring by assigning cluster number to each data object
        based onconnection
Step 6: Obtain the modularity of the clusters obtained.
Step 7: Perform mutation by reassigning the cluster numbers based on the
        significance of the edge connection by referring the edge matrix
Step 8: Iterate steps 6-7 till the maximum modularity is obtained
```

Process flow of the proposed algorithm is shown in Fig. 1 and is detailed in the following steps using a simulated dataset.

Step 1 (edge pruning process): Primarily, the edges are sorted by their weights in ascending sequence. The best path is then identified using Dijkstras algorithm. For each iteration, check whether the selected edges are a part of best path. If the edges form a best path, the next edge will be selected and the edges are pruned only if it is not part of the best path. Figure 2a and b represents a fully

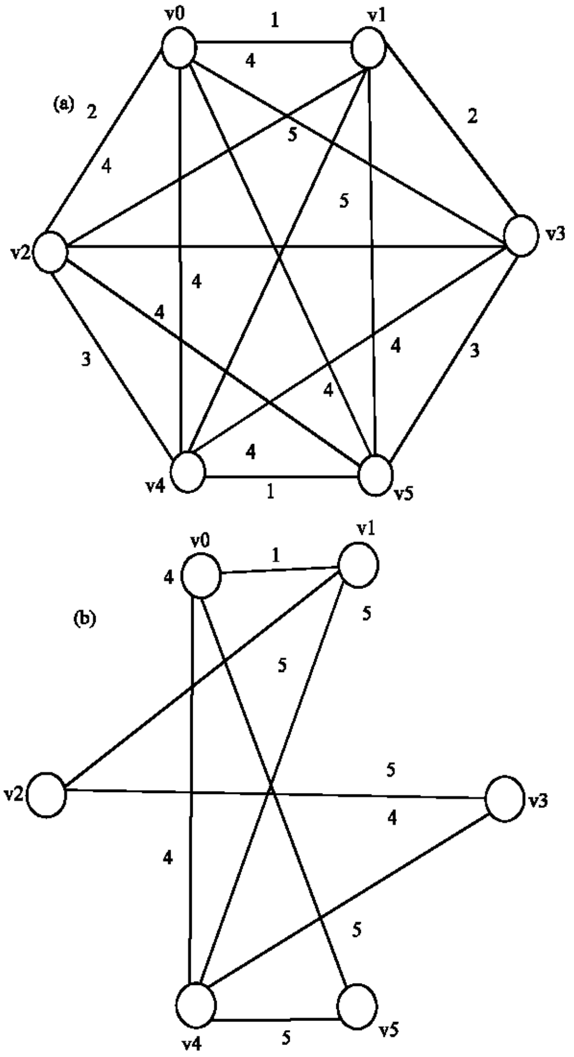


Fig. 2: a) Rrepresents sample complex graph and b) gives the resultant pruned graph

Closeness		Degree	
V0	7	V0	3
V1	7	V1	3
V2	9	V2	2
V3	8	V3	2
V4	6	V4	4
V5	9	V5	2

Fig. 3: The connectivity matrix of each node based on centrality measures-"closeness" and "degree"

Table 1: Edge matrix of pruned graph

Network	V0	V1	V2	V3	V4	V5
V0		*			*	*
V1	*		*		*	
V2		*		*		
V3			*		*	
V4	*	*		*		*
V5	*				*	

connected graph and Fig. 2b represents the resultant pruned graph. The optimum network derived can be transformed to an Edge matrix as shown in Table 1.

Step 2 (building solution space based on centrality measures): Next step is to derive meaningful clusters from the simplified network (Edge matrix). An N*1 connectivity matrix is formed based on network centrality measures like "closeness" and "degree". This is used as initial solution space. Figure 3 shows the solution space using closeness and degree centrality.

Degree centrality: Degree centrality can be defined as the number of links incident into a node.

Closeness centrality: Closeness centrality is the average length of the shortest path between the nodes and all other nodes in the network (Elsner, 1997).

The connectivity matrices are then sorted in descending order. Solution space (offspring) is then generated from this matrix by assigning cluster ID (1-k) for each data objects; "1" for the first node and all the nodes that are connected to this node based on edge matrix. The process repeats till all the nodes are assigned with the cluster number. Figure 4 represents the sorted matrix and the solution generated from that matrix.

Step 3 (optimizing the clusters based on mutation operator): The optimization technique used for detecting the community structure in network is to maximize the modularity value using the equation:

Closeness		Degree	
V5	9	V5	1
V2	9	V2	2
V3	8	V3	2
V0	7	V0	1
V1	7	V1	2
V4	6	V4	1

Fig. 4: Sorted matrix based on closeness parameter and the offspring generated based on the sorted matrix

Table 2: Clusters derived initially

Cluster 1			Cluster 2		
V5	V0	V4	V2	V3	V1
1a	1b	1c	2a	2b	2c

Table 3: Clusters after first mutation

Cluster 1			Cluster 2		
V1	V3	V4	V2	V0	V5
1a	1b	1c	2a	2b	2c

Table 4: Clusters after second mutation

Cluster 1			Cluster 2		
V1	V2	V4	V3	V0	V5
1a	1b	1c	2a	2b	2c

$$F(x) = \max(Q), \quad Q = \sum_{i=1}^{k}\left[e_{ii} - a_i^{2}\right]$$

Where:

e_{ii} = The probability of edge in module (cluster)

i and a_i = The probability of a random edge would fall into module (cluster) i

Modularity is designed to measure the strength of each modules or clusters. The value of modularity is in the range [-0.5, 1]. Clusters with high modularity has strong connection between the nodes within the modules and sparse connections between nodes in different modules. Here, mutation operator is used to maintain the diversity in the population and to avoid local maxima. There are different types of mutation such as insert mutation, inversion mutation, scramble mutation, swap mutation, Flip mutation, etc. (Sreeja and Krishnakumar, 2017). Here, we use mutation because it selects two alleles and swaps its position by maintaining the adjacency information.The cluster IDs are reassigned based on the significance of the connection recorded in the edge matrix (Hadj and Belbachir, 2014). Table 2-4 represent different iteration of mutation operation and the generation of clusters in each phase. The optimal solution is derived based on the clusters showing maximum modularity value. The modularity of the above network before mutation is 0.106, after the first mutation it is 0.374 and then for second mutation again its incremented to 0.413. Based on, the number of iterations set, the process can be repeated and extract the optimum clusters showing maximum modularity.

RESULTS AND DISCUSSION

Experiment and result analysis: The algorithm is implemented in R. The analysis was done using yeast dataset taken from UCI repository, a research data repository for machine learning and intelligent systems (Lumley, 2011). A total of 503 data instances were extracted for the study. Each instance is explained by 9 attributes. Both closeness and degree measures are used to check the performance of the clusters. Figure 5a represents the modularity values of clusters derived using closeness measure and Fig. 5b represents the clusters derived using degree measure. It is observed that the clusters derived using degree measure is showing better performance than the closeness parameter. The optimum value is obtained with a modularity equal to 0.2475416.

The internal validation measure Silhouette index (Wang *et al*., 2009) is also computed for the optimum clusters derived and is also compared with standard k-means clustering algorithm with cluster size, k = 4. The clusters derived using proposed approach shows a value of 0.739 whereas clusters obtained using k-means shows only a value of 0.518. This clearly emphasizes the significance of the proposed approach for analyzing complex biological data.

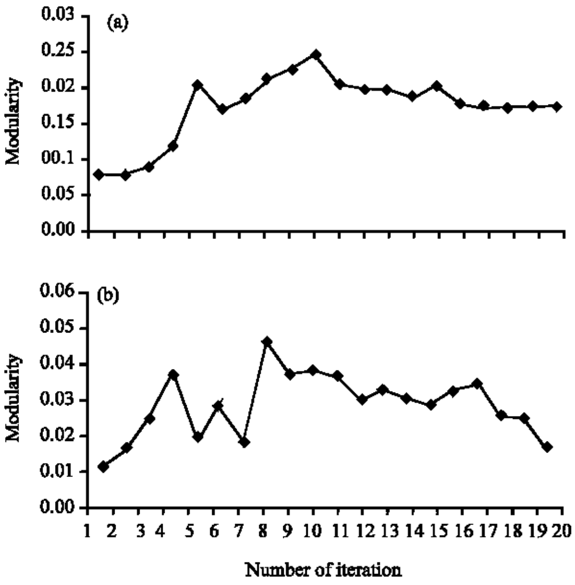


Fig. 5: Modularity values of the clusters derived using: a) Degree and b) Closeness measures

CONCLUSION

Understanding, analysing and visualizing complex biological data requires several pre-processing steps and data mining approaches. Graph partitioning approaches are efficient techniques to uncover hidden knowledge for in-depth analysis from the huge heterogeneous information. But the complexity of the analysis increases when we process the fully connected network. In the present work, simplification of complex networks using edge pruning techniques are explored and proposed an optimized clustering process for deriving meaningful clusters using Genetic algorithm from the pruned network. It has been observed that the modularity has been improved in each iteration and has reached an optimum level for the clusters derived using degree measure. The silhouette index of the optimum clusters also shows better performance when compared with the standard k-means algorithm. This shows that the identification of significant edges using edge pruning technique act as an important pre-processing process for building optimum clusters. The use of mutation operator to fine tune the positions of the data objects also enhances the performance of the clustering process.

ACKNOWLEDGEMENT

This research is supported by the DST Funded Project, (SR/CSI/81/2011) under Cognitive Science Research Initiative in the Department of Computer Science, School of Arts and Sciences, Amrita Vishwa Vidyapeetham, Kochi.

REFERENCES

Albert, R. and A.L. Barabasi, 2002. Statistical mechanics of complex networks. Rev. Modern Phys., 74: 47-97.

Alexandrescu, A., C. Mitica and A. Ioan, 2011. Determining the best mutation probabilities of a Genetic algorithm for mapping tasks. Buletinul Institutului Politehnic din Iasi, Iowa. http://www.ace.tuiasi.ro /users/103/f2 _2011 _2_(21 _30)Alexandrescu.pdf

Ashok, S. and M. Judy, 2016. Exploring key gene interactions using particle swarm optimization. Intl. J. Pharma Bio Sci., 7: 734-741.

Ashok, S. and M.V. Judy, 2015. A novel iterative partitioning approach for building prime clusters. Intl. J. Adv. Intell. Paradigms, 7: 313-325.

Biedl, T.C., B. Brejova and T. Vinar, 2000. Simplifying flow networks. In: Proceedings of the International Symposium on Mathematical Foundations of Computer Science, Nielsen, M. and B. Rovan (Eds.). Springer, Berlin, Germany, pp: 192-201.

Bodenhofer, U., 2003. Genetic Algorithms: Theory and Applications. 3rd Edn., Fuzzy Logic Laboratorium Linz Hagenberg, Linz, Austria, Pages: 126.

Chapela, V., R. Criado, S. Moral and M. Romance, 2015. Mathematical Foundations: Complex Networks and Graphs (A Review). In: Intentional Risk Management Through Complex Networks Analysis, Victor, C., C. Regino, M. Santiago and R. Miguel (Eds.). Springer, Berlin, Germany, ISBN:978-3-319-26421-9, pp: 9-36.

Elsner, U., 1997. Graph Partitioning: A Survey. Chemnitz University of Technology, Chemnitz, Germany, Pages: 58.

Enright, J. and M. Kitty, 2015. Deleting edges to restrict the size of an epidemic: A new application for treewidth. Prceedings of the 9th International Conference on Combinatorial Optimization and Applications, December 18-20, 2015, Springer, Houston, Texas, ISBN:978-3-319-26625-1, pp: 574-585.

Eubank, S., H. Guclu, V.S.A. Kumar, M.V. Marathe, A. Srinivasan, Z. Toroczkai and N. Wang, 2004. Modeling disease outbreaks in realistic urban social networks. Nature, 429: 180-184.

Hadj, T.K. and H. Belbachir, 2014. Comparative study of quality measures of sequential rules for the clustering of web data. Models Optimisation Math. Anal. J., 2: 29-35.

Leena, A.M., P. Surya, A. Sreeja and M.V. Judy, 2016. Application of ant colony optimization in identifying the key gene interactions.q Intl. J. Control Theory Appl., 9: 4211-4219.

Li, Z. and J. Liu, 2016. A multi-agent genetic algorithm for community detection in complex networks. Phys. A: Stat. Mech. Appl., 449: 336-347.

Liu, G. and W. Limsoon, 2008. Effective pruning techniques for mining quasi-cliques. Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases, September 15-19, 2008, Springer, Antwerp, Belgium, pp: 33-49.

Lumley, T., 2011. Complex Surveys: A Guide to Analysis using R. John Wiley & Sons, Hoboken, New Jersey, ISBN:978-0-470-28430-8, Pages: 67.

Ma, K.L., 1999. Image graphs: A novel approach to visual data exploration. Proceedings of the Conference on Visualization'99 Celebrating Ten Years, October 22, 1999, IEEE, San Francisco, California, ISBN:0-7803-5897-X, pp: 81-88.

Misiolek, E. and D.Z. Chen, 2006. Two flow network simplification algorithms. Inf. Process. Lett., 97: 197-202.

Pramanik, S., U.N. Chowdhury, B.K. Pramanik and N. Huda, 2010. A comparative study of bagging, boosting and C4.5: The recent improvements in decision tree learning algorithm. Asian J. Inform. Technol., 9: 300-306.

Soni, N. and T. Kumar, 2014. Study of various mutation operators in genetic algorithms. Int. J. Comput. Sci. Inform. Technol., 5: 4519-4521.

Sreeja, A. and U. Krishnakumar, 2017. Gene ontology based functional analysis and graph theory for partitioning gene interaction networks. Intl. J. Pharma Bio Sci., 8: 183-192.

Sumathi-Punitha, C.P. and T. Santhanam, 2007. A combination of genetic algorithm and ART neural network for breast cancer diagnosis. Asia J. Inform. Technol., 6: 112-117.

Toivonen, H., S. Mahler and F. Zhou, 2010. A framework for path-oriented network simplification. Proceedings of the 9th International Symposium on Intelligent Data Analysis (IDA 2010), Springer, Tucson, Arizona, May 19-21, 2010, pp: 220-231.

Toussaint, G.T., 1980. The relative neighbourhood graph of a finite planar set. Pattern Recognit., 12: 261-268.

Wang, K., B. Wang and L. Peng, 2009. CVAP: Validation for cluster analyses. Data Sci. J., 8: 88-93.

Zhou, F., 2012. Methods for Network Abstraction. University of Helsinki, Helsinki, Finland, ISBN:978-952-10-8157-6, Pages: 56.

Zhou, F., S. Mahler and H. Toivonen, 2012. Simplification of Networks by Edge Pruning. In: Bisociative Knowledge Discovery, Michael, R.B. (Ed.). Springer, Berlin, Germany, ISBN:978-3-642-31829-0, pp: 179-198.