Journal of Nature, Science & Technology 4 (2021) 16-18
Nature, Science & Technology
www.acapublishing.com

Research Article

Study of Gene-Gene Interaction Networks

Prerna Saurabh*[,1], Amar Kumar Verma[2], Saurabh Rai[3], Sourabh Singla[4]
[1]*Vellore Institute of Technology, Vellore Campus, Tamilnadu*
[2]*Birla Institute of Technology and Science, Pilani- Hyderabad campus, Telangana*
[3]*Indian Institute of Technology, Ropar, Punjab*
[4]*Defence Institute of Advanced Technology, Maharashtra*

Keywords

Gene-Gene interaction, Driver nodes, Statistical analysis, DNA, Genetic, k-nearest neighbors *(KNN)*

Abstract

Gene-Gene Interactions (GGI) Networks-Genomics essentially finds the driver node to understand the functional mechanism of Gene-Gene Interaction or GGI. This process can significantly improvise while examining molecular processes perturbed by genetics in human diseases. With Artificial Intelligence (AI) developments, pattern recognition and machine learning advancements can be exploited to automate from more superficial to handle any task in the medical field. In the past, deep learning-based methods have provided encouraging results in the medical field, such as breast cancer detection, skin cancer classification, brain disease classification, arrhythmia detection, pneumonia detection from X-Ray images, and lung segmentation. Interestingly, employing machine learning in the medical field has caught my great attention and motivated us to utilize machine learning-based algorithms to detect drive nodes. Several recent types of research have focused on identifying the bare minimum of driver nodes required to control underlying gene-gene interaction networks. This study is about analyzing gene interaction networks statistically. One or more abnormalities cause a genetic condition in the DNA. Disease-related genes play essential biological roles in the cell. Multiple genes often work together to cause complex genetic illnesses. So, the central concept is to create a system that can assess networks in various elements that influence them.

1.Introduction

A gene interaction network collects genes (nodes) linked by edges that reflect functional connections between them [1]. Because the two provided genes are considered either a physical connection through their gene products, such as proteins, or one of the genes changes or impacts the activity of another gene of interest, these edges are called interactions. Genes and their functional products, such as proteins, collaborate to complete a task. They frequently create physical bonds with one another to operate or build a more complicated structure. These interactions can last a long time, such as when proteins create protein complexes, or they can be short, such as when proteins alter each other [2]. Besides these physical interactions, there are genetic interactions in which two gene variants have a combined effect that does not manifest itself with only one of them alone. A genetic disease is caused by one or more defects in the DNA. Disease-related genes have essential biological roles in the cell. Multiple genes typically work together to cause complex genetic diseases. Consequently, the route phenotype is caused by disruptions in the underlying pathways, in which genes collaborate via the different mechanisms mentioned previously. Gene interactions are therefore critical for understanding how genes interact in model organisms and for gaining insight into complicated illnesses. The driver nodes are those that allow us to control the whole network. To manage the entire gene interaction network, a crucial set of genes is necessary. They have high betweenness centrality. So, they control information flow in a network and hence may be subject to a targeted attack. So, to get a better understanding of GGI, determining driver nodes is crucial.

2.Related Works

Genes, which are essential macromolecules, have been proven to function on their own in studies. Genes interact physically with other partners to carry out a variety of molecular activities within a cell. As a result, Gene-Gene Interactions (GGI) is an essential tool for understanding the cell's structural and functional architecture. Many GGIs have been created and gathered due to the development of high-throughput methods, paving the path for the formation of GGI networks. Identifying the driver Gene has become a critical problem in systems biology to understand the working mechanism of GGI networks better. The driver nodes are the nodes that allow us to govern the whole network. The MDS (Minimum Dominating Set) model is frequently used in general. The MDS model, on the other hand, does not provide a specific MDS configuration. Even yet, numerous formats are created if various optimization methods are used, making it impossible to establish the real set of driver nodes. The CC-MDS model (centrality-corrected minimum dominating stage) was created and contains two crucial factors: To discover the real set of driver nodes in a particular GGI network, use 1) Degree Centrality and 2) Betweenness Centrality. The CCMDS model focuses on genes with a high degree of centrality and a high degree of betweenness. In contrast, the MDS model focuses on genes with a low degree of centrality. Because of its more central location, CC-MDS Gene is more important than MDS Gene in ensuring overall network connection. To demonstrate the functional importance, we discovered that CC-MDS Gene is engaged in more gene complexes and GO annotations on average than MDS Gene.

We also discovered that CC-MDS Gene contains more important genes, aging, disease-associated, and virus-targeted genes, than MDS Gene. When it comes to transcription factors and gene kinases, the CC-MDS Gene sets have a considerably higher enrichment of transcription factors and gene kinases. The topological and functional significance results show that the CC-MDS model can capture driver genes than the MDS model. According to the centrality-lethality criterion, the GGI network's strongly connected Gene is more important. After that, the

*Corresponding Author:
prerna.saurabh2019@vitstudent.ac.in
(Prerna Saurabh Orcid: 0000-0001-9665-767X)
Received 15 June 2021 Revised 28 Aug 2021 Accepted 28 Aug 2021
Journal of Nature, Science & Technology 4 (2021) 16-18
https://doi.org/10.36937/janset.2021.004.004
2757-7783 © 2019 ACA Publishing. All rights reserved.

relationship between degree and essentiality was verified, and its reasons were investigated. Unlike degree centrality, which counts how many neighbors a node has, betweenness centrality measures how many shortest routes cross through it. The "information transfer" is significantly influenced by a node with a high betweenness centrality. As a result, high-betweenness genes may serve as critical network connections.

According to studies, the centrality index in a GGI network may also be a valuable measure of its biological and functional relevance. However, as far as we know, no systematic attempt has been made to determine whether high-degree genes or high-betweenness genes can provide total control over the underlying network. If a single connection can access all the remaining (i.e., non-DS) nodes, the network set is called a dominant set (DS). The MDS is thus defined as the network's smallest DS (see Figure 1).

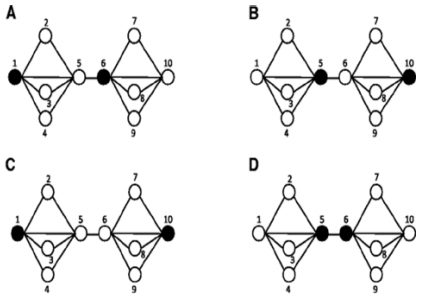

Figure 1. The methods employed in determining the MDS and CC-MDS model are followed with an example.

Milenkovi et al. devised two heuristic methods to discover dominant sets in GGI networks inspired by telecommunications applications. They found that dominant groups had a higher proportion of physiologically essential genes. Their approaches, however, may not generate minimum dominant groups. According to studies, the projected driver genes in the MDS model have crucial functional features (e.g., essential genes, cancer-related genes, and virus-targeted genes) and play a vital role in driving the whole network, according to studies [3]. Various optimization techniques used to solve the MDS model, on the other hand, may produce different configurations (see Figure 1. A-D). As a result, the MDS model does not generate a unique collection of driver genes. Nacher and Akutsu used Jia et al. techniques to divide the nodes into three categories:

- Crucial nodes, which belong to every configuration.
- Redundant nodes, which never belong to any structure.
- Intermittent nodes, which belong to some but not all structures.

In this process, the MDS model must be solved n times, where n is the number of nodes. As a result, compared to generating an MDS, their technique necessitates much more CPU time. In their study, the biological relevance of the three types of nodes is not examined. Please remember that the described approaches may also be utilized to compute crucial nodes in directed networks. The MDS model has recently been significantly expanded and implemented. It was created, for example, to deal with the controllability of bipartite networks. Based on the MDS model's framework.

Nacher and Akutsu introduced structurally robust control of complex networks where at least two nodes must cover each node in the dominating set. This section considers a different extension of the standard MDS model to include heterogeneity in the degree and the betweenness of genes. We develop a degree and betweenness centrality-corrected MDS (CC-MDS) model based on the assumption that high-degree and high-betweenness genes are more likely to be controllers, we develop a degree and betweenness centrality-corrected MDS (CC-MDS) model [4]. Despite its innocuous appearance, this corrected version turns out to have substantial effects. Researchers of the assigned paper had run both the standard MDS model and the CC-MDS model on three human GGI networks. Experiment results show that CC-MDS genes (driver genes determined using the CC-MDS model) predicted by different optimization methods are almost identical, while the overlap between MDS genes (driver genes selected using the MDS model) is expected by other optimization methods is relatively low. We also observe that CC-MDS genes are more important in maintaining the overall network connectivity than MDS genes.

2. Methodology

2.1. Approach to find a driver node

Step 1: We take an adjacency matrix as the input.
Step 2: For each pair of vertices (contains genes), node centrality is found.
Step 3: Each node centrality is stored in a dictionary or hash table and its corresponding node.
Step 4: Nodes having higher node centrality values than the threshold can be concluded as novel driver nodes.

The fact that the spread scores for most of the analyzed real-world networks fluctuate primarily in the sub-interval [0.3, 0.6] is evident. However, the spread-scores for various metrics are neither obvious or comparable. Therefore, the simulation results for threshold values of 0.3, 0.4, 0.5, and 0.6, i.e., thresholds in percentages of 30%, 40 %, 50%, and 60% [5].

2.2. Algorithm for similar community detection

This algorithm uses centrality betweenness or Edge betweenness-based community detection. (Detection of similar novel driver nodes). Here we opt for Girvan Newman Algorithm. The algorithm works as follows:

Step 1: calculate the centrality betweenness between all the edges (and store it in a dictionary or hash table preferred).
Step 2: Remove the edge having the highest betweenness centrality.
Step 3: Repeat Steps 1 and 2 until a certain threshold is achieved.

2.3. Analysis of the algorithm

1. Fastest version of the Girvan Newman Algorithm to calculate centrality betweenness is O(m*n), where m is no. of edges and n is no. of vertices.
2. Due to its repeating nature, sometimes the complexity can be O(m*m*n).
3. This algorithm can be optimized using the modularity concept.

Inference:
This algorithm helps us to analysis the extent of clustering in the network.

4. Checking spread of driver nodes via Machine Learning

We decide to train a model via machine learning to give all k-nearest driver nodes in the network. For this model, we choose to go for Instance-Based Learning is also known as a Lazy Algorithm. In this algorithm, the algorithm does not come up with the model. Instead, it checks and uses the stored instances in the memory to get the output in testing instances.

4.1 K-Nearest Neighbor

Training phase: save the example.

Prediction phase: Get the test instance X(t). Find in training example (x1, y1) such that it is closest to X(t), then predict y1 to be the output as Y(t). This is for the single nearest neighbor. Similarly, for k-neighbors, find 'k.' training example ((x1, y1), (x2, y2), (x3, y3), (xk,yk)) such that it is closest to X(t) then predict y1 to be the output as Y(t).

Classification: Predict the majority class from {y1, y2, y3, yk}.

Regression: Predict the average from {y1, y2, y3, yk}.

The degree, betweenness and weight for given network in Figure 1 is listed in Table 1. MDS concerning original and corrected model using k-nn is listed in Table 2.

Table 1. The degree, betweenness and weight for given network in Figure 1.

Node	Degree	Betweenness	Weight (y=0.05)
1	4	0.0833	1.0565
5	5	1.1944	0.9145
6	5	1.1944	0.9145
10	4	0.0833	1.0565

Table 2. MDS concerning original and corrected model using k-nn.

MDS	Original Model	Corrected Model	Percentage Error
A {1,6}	2	1.971	1.471
B {5,10}	2	1.971	1.471
6 {1,10}	2	2.113	-5.347
10 {5,6}	2	1.829	9.349

A few other aspects of extracting driver nodes from gene-gene interactions with Artificial Intelligence-based approach can be made. With advancements in Artificial Intelligence Techniques, the machine learning algorithms can be exploited in several applications. Various ML approach for different applications have been extensively used in [6-15].

5. Conclusions

In the proposed system, we aim to deal with the statistical analysis of the Gene-Gene Interaction dataset of human cancer. We have solely been focusing on the driving nodes in a sample of clustered genomes. The system will provide the following calculations:

i. Node count

ii. Network diameter

iii. Clustering Coefficient

iv. Network centralization

v. Network Density

vi. Mean

All these calculations have been done on a graph that will be consisting of normal as well as driver nodes.

Declaration of Conflict of Interests

The authors declare that there is no conflict of interest. They have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1.] Barabasi A-L, Oltvai ZN, Network biology: understanding the cell's functional organization. Nat Rev Genet 5(2004)101–13.

[2.] Vinayagam A, Zirin J, Roesel C, Hu Y, Yilmazel B, Samsonova AA, integrating protein-protein interaction networks with phenotypes reveal signs of interactions. Nat Methods 11(2013) 94–9.

[3.] Wuchty S, Controllability in protein interaction networks. Proc Nat Acad Sci USA 111(2014) 7156–760.

[4.] Freeman, Linton C., A set of measures of centrality based on betweenness. Sociometry (1977) 35-41.

[5.] Singh, Anuj, Rishi Ranjan Singh, and S. R. S. Iyengar. "Node-weighted centrality: a new way of centrality hybridization." Computational Social Networks 7, no. 1 (2020): 1-33.

[6.] Amar Kumar Verma, Sudha Radhika, and Naren Surampudi, Web based application for quick and handy health condition monitoring system for a reliable wind power generation. In ASME International Mechanical Engineering Congress and Exposition 84669 (2020) V014T14A009.

[7.] GSK Ranjan, Amar Kumar Verma, and Sudha Radhika, K-nearest neighbors and grid search cv based real time fault monitoring system for industries. International conference for convergence in technology (2019) 1-5.

[8.] Inala Vivek Vamsi, Nippani Abhinav, Amar Kumar Verma, and Sudha Radhika, Random Forest based real time fault monitoring system for industries. International Conference on Computing Communication and Automation (2018)1-6.

[9.] Amar Kumar Verma, Pragnya Akkulu, Shravvan V Padmanabhan, and Sudha Radhika, Automatic condition monitoring of industrial machines using fsa-based hall-effect transducer. IEEE Sensors Journal 21(2020) 1072-1081.

[10.] Amar Kumar Verma, Aakruti Jain, and Sudha Radhika, Neuro-fuzzy classifier for identification of stator winding inter-turn fault for industrial machine. In International conference on Modelling Simulation and Intelligent Computing (2020) 101-110.

[11.] Amar Kumar Verma, Shivika Nagpal, Aditya Desai, and Radhika Sudha, An efficient neural-network model for real-time fault detection in industrial machine. Neural Computing and Applications 33 (2021) 1297-1310.

[12.] Amar Kumar Verma, Sudha Radhika, and SV Padmanabhan, Wavelet based fault detection and diagnosis using online mcsa of stator winding faults due to insulation failure in industrial induction machine. Recent Advances in Intelligent Computational Systems (2018) 204-208.

[13.] Amar Kumar Verma, P Spandana, SV Padmanabhan, and Sudha Radhika, Quantitative modeling and simulation for stator inter-turn fault detection in industrial machine. International conference on intelligent computing and communication (2019) 87-97.

[14.] Amar Kumar Verma, Jaju Vedant Vinod, and Radhika Sudha, A modular zigbee-based iot platform for reliable health monitoring of industrial machines using refsa. Microelectronics and Signal Processing (2021) 179-188.

[15.] Verma, Amar Kumar, Inturi Vamsi, Prerna Saurabh, Radhika Sudha, G. R. Sabareesh, and S. Rajkumar. "Wavelet and deep learning-based detection of SARS-nCoV from thoracic X-ray images for rapid and efficient testing." Expert Systems with Applications (2021): 115650.

How to Cite This Article

Saurabh, P., Verma, A.K., Rai, S:, Singla, S., Study of Gene-Gene Interaction Networks, Journal of Nature, Science & Technology, 4 (2021), 16-18. https://doi.org/10.36937/janset.2021.004.004